LEGAL KNOWLEDGE AND INFORMATION SYSTEMS

VISIT…

LANZAROTE
Caliente.com

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including "Information Modelling and Knowledge Bases" and "Knowledge-Based Intelligent Engineering Systems". It also includes the biannual ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Volume 134

Recently published in this series

Vol. 133. C.-K. Looi et al. (Eds.), Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences – Sharing Good Practices of Research, Experimentation and Innovation
Vol. 132. K. Nakamatsu and J.M. Abe (Eds.), Advances in Logic Based Intelligent Systems – Selected Papers of LAPTEC 2005
Vol. 131. B. López et al. (Eds.), Artificial Intelligence Research and Development
Vol. 130. K. Zieliński and T. Szmuc (Eds.), Software Engineering: Evolution and Emerging Technologies
Vol. 129. H. Fujita and M. Mejri (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the fourth SoMeT_W05
Vol. 128. J. Zhou et al. (Eds.), Applied Public Key Infrastructure – 4th International Workshop: IWAP 2005
Vol. 127. P. Ritrovato et al. (Eds.), Towards the Learning Grid – Advances in Human Learning Services
Vol. 126. J. Cruz, Constraint Reasoning for Differential Models
Vol. 125. C.-K. Looi et al. (Eds.), Artificial Intelligence in Education – Supporting Learning through Intelligent and Socially Informed Technology
Vol. 124. T. Washio et al. (Eds.), Advances in Mining Graphs, Trees and Sequences
Vol. 123. P. Buitelaar et al. (Eds.), Ontology Learning from Text: Methods, Evaluation and Applications

ISSN 0922-6389
Frontiers in Artificial Intelligence and Applications

ISSN 1570-3886
Legal Knowledge and Information Systems

Legal Knowledge and Information Systems

JURIX 2005: The Eighteenth Annual Conference

Edited by

Marie-Francine Moens

Katholieke Universiteit Leuven, Belgium

and

Peter Spyns

Vrije Universiteit Brussel, Belgium

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 1-58603-576-2
Library of Congress Control Number: 2005909820

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
IOS Press/Lavis Marketing
73 Lime Walk
Headington
Oxford OX3 7AD
England
fax: +44 1865 750079

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

Preface

This volume contains the Proceedings of the Eighteenth JURIX Conference on Legal Knowledge and Information Systems (JURIX 2005), which is held on December 8–10, 2005 at the Vrije Universiteit Brussel in Brussels, Belgium.

Thirteen full papers and seven extended abstracts are included in these Proceedings. Authors, invited speakers and workshop organizers come from Australia, Austria, Belgium, Canada, China, France, Germany, Italy, Portugal, Russia, Spain, Switzerland, United Kingdom, and the United States of America. It seems that the JURIX conference is becoming the leading European conference on legal informatics and artificial intelligence and law.

A number of papers discuss traditional topics of artificial intelligence and law and draft models of law and legal reasoning. The research of Katie Atkinson and Trevor Bench-Capon is a valuable attempt to integrate different existing models of legal reasoning. Guido Governatori et al. discuss a framework for defeasible reasoning. Alexander Boer, Tom van Engers and Radboud Winkels argue that legal norms are in many contexts best understood as expressions of a ceteris paribus preference, and that this viewpoint adequately accounts for normative conflict and contrary-to-duty norms. The paper of Christopher Giblin et al. introduces a meta-model and method for modeling regulations and managing them in a systematic lifecycle in an enterprise. Jeroen Keppens and Burkhard Schafer discuss evidentiary reasoning and its formalization in a first-order assumption-based reasoning architecture. Moshe Looks, Ronald P. Loui and Barry Z. Cynamon present a mathematical modeling method of agents that pursue their interests and of a legislator who tries to influence the agents in ways that promote the legislator's goals.

Three papers are on the topic of legal knowledge acquisition using natural language processing. Pietro Mercatali et al. discuss the first steps that are needed for the automatic translation of textual representations of laws into formal models. The paper of Farida Aouladomar analyzes the form, presentation, meaning and modes of answering procedural questions ("how") in the context of online e-Government applications. Paolo Quaresma and Irene Pimenta Rodrigues discuss a question answering system for legal information retrieval.

A number of short papers describe very interesting work in progress and often focus on practical applications such as reducing the legal burden, planning a new bill, classification of legislative documents, and reasoning tools for e-Democracy.

A final section of the Proceedings is devoted to the use of ontologies in describing the law. The paper of Ronny van Laarschot et al. attempts to bridge the gap between a laymen's description and legal terminology. Peter Spyns and Giles Hogben apply and validate an automatic evaluation procedure on ontology mining results from the EU privacy directive. Roberto García and Jaime Delgado present an ontological approach for the management of data dictionaries of intellectual property rights. Finally, Laurens Mommers and Wim Voermans explain how cross-lingual information retrieval is useful in the legal field.

Invited lectures were given by Luc Wintgens and Helmut Horacek.

This conference focuses on two major themes and their integration: Artificial Intelligence and e-government. Currently, several artificial intelligence technologies are growing increasingly mature, including computational modeling of reasoning, natural language processing, information retrieval, information extraction, machine learning, electronic agents, and reasoning with uncertainty. Their integration in and adaptation to legal knowledge and information systems need to be studied. Parallel to this development, e-government applications are gradually gaining ground among local, national, European and international institutions. More than 25 years of research in the field of legal knowledge and information systems have resulted in many models for legal knowledge representation and reasoning. However, authors and reviewers rightly remarked that there are still some essential questions to be solved. First, there is a need for the integration and harmonization of the models. Secondly, there is the difficult problem of knowledge acquisition in a domain that is in constant evolution. If one wants to realize a fruitful marriage between artificial intelligence and e-government, the aid of technologies that automatically extract knowledge from natural language and from other forms of human communication and perception is needed.

The organizing committee of JURIX 2005 consists of Peter Spyns, Greet Janssens, Johan Verdoodt, Pieter De Leenheer and Yan Tang. This committee is very grateful to Koen Deschacht, Toon Lenaerts and Roxana Angheluta for their extra help. We especially thank the members of the program committee of this conference:

- Jon Bing, University of Oslo, Norway
- Kevin D. Ashley, University of Pittsburgh, USA
- Trevor J.M. Bench-Capon, University of Liverpool, UK
- Pascale Berteloot, Office des publications, European Commission
- Karl Branting, BAE systems, USA
- Jaime Delgado, Universitat Pompeu Fabra, Spain
- Aldo Gangemi, Institute of Cognitive Sciences and Technologies, Italy
- Thomas F. Gordon, Fraunhofer FOKUS, Berlin, Germany
- Eduard Hovy, University of Southern California, USA
- Ronald Leenes, Tilburg University, The Netherlands
- Richard Leary, University College London, UK
- Arno Lodder, Vrije Universiteit Amsterdam, The Netherlands
- Anja Oskamp, Vrije Universiteit Amsterdam, The Netherlands
- Henry Prakken, Utrecht University/University of Groningen, The Netherlands
- Giovanni Sartor, Università di Bologna, Italy
- Erich Schweighofer, University of Vienna, Austria
- Peter Spyns, Vrije Universiteit Brussel, Belgium
- Roland Traunmüller, University of Linz, Austria
- Tom van Engers, University of Amsterdam, The Netherlands
- Bart Verheij, Rijksuniversiteit Groningen, The Netherlands
- Radboud Winkels, University of Amsterdam, The Netherlands
- John Zeleznikow, Victoria University, Australia

Leuven, October 26, 2005
Marie-Francine Moens, Chair of the program committee
Peter Spyns, Chair of the organizing committee

Contents

Preface v
Marie-Francine Moens and Peter Spyns

Theory and Practice in AI and Law: A Response to Branting 1
Katie Atkinson and Trevor Bench-Capon

Norm Modifications in Defeasible Logic 13
Guido Governatori, Monica Palmirani, Regis Riveret, Antonio Rotolo and Giovanni Sartor

Making Sense of Coherence: The Level Theory of Coherence 23
Luc J. Wintgens

Mixing Legal and Non-Legal Norms 25
Alexander Boer, Tom van Engers and Radboud Winkels

Regulations Expressed as Logical Models (REALM) 37
Christopher Giblin, Alice Y. Liu, Samuel Müller, Birgit Pfitzmann and Xin Zhou

Assumption Based Peg Unification for Crime Scenario Modelling 49
Jeroen Keppens and Burkhard Schafer

Dynamics of Rule Revision and Strategy Revision in Legislative Games 59
Moshe Looks, Ronald P. Loui and Barry Z. Cynamon

Automated Analysis of Reasoning and Argumentation Structures in Texts 69
Helmut Horacek

Automatic Translation from Textual Representations of Laws to Formal Models Through UML 71
Pietro Mercatali, Francesco Romano, Luciano Boschi and Emilio Spinicci

Some Foundational Linguistic Elements for QA Systems: An Application to E-Government Services 81
Farida Aouladomar

A Question Answer System for Legal Information Retrieval 91
Paulo Quaresma and Irene Pimenta Rodrigues

How Technology Can Help Reducing the Legal Burden 101
Tom van Engers, Ron van Gog and Arian Jacobs

A Semantics-Based Visual Framework for Planning a New Bill 103
Carlo Biagioli and Enrico Francesconi

The Desocialization of the Courts, Sentencing Decision Support and Plea Bargaining 105
Andrew Vincent and John Zeleznikow

Towards Modeling Systematic Interpretation of Codified Law 107
Matthias Grabmair and Kevin D. Ashley

Large-Scale Linguistic Ontology as a Basis for Text Categorization of Legislative Documents 109
Natalia Loukachevitch and Boris Dobrov

Game Mechanisms & Procedural Fairness 111
Moshe Looks and Ronald P. Loui

Practical Reasoning and Proposing: Tools for e-Democracy 113
Douglas Walton

The Legal Concepts and the Layman's Terms 115
Ronny van Laarschot, Wouter van Steenbergen, Heiner Stuckenschmidt, Arno R. Lodder and Frank van Harmelen

Validating an Automated Evaluation Procedure for Ontology Triples in the Privacy Domain 127
Peter Spyns and Giles Hogben

An Ontological Approach for the Management of Rights Data Dictionaries 137
Roberto García and Jaime Delgado

Using Legal Definitions to Increase the Accessibility of Legal Documents 147
Laurens Mommers and Wim Voermans

Previous Publications by JURIX 157

Author Index 161

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Theory and Practice in AI and Law: A Response to Branting

Katie Atkinson and Trevor Bench-Capon
Department of Computer Science
University of Liverpool
Liverpool L69 7ZF, UK
{katie,tbc}@csc.liv.ac.uk

Abstract. In this paper we use our previous work which has examined the different levels involved in reasoning about legal cases to examine some challenges to the relevance of current theoretical work in AI and Law made by Branting. In our model we view the process of legal reasoning as being divided into three distinct but interconnected levels of reasoning. These levels involve a bottom layer concerning facts about the world, a top layer concerning legal consequences, and a layer connecting the two, with conclusions at lower levels acting as premises for higher levels. We use our model to explain Branting's observations and show the relation with other strands of work from the AI and Law community.

Keywords. Legal Reasoning, Legal Information Systems, Theory Construction

1. Introduction

One way in which Artificial Intelligence progresses is through a consideration of specific problems. Such problems provide benchmarks which can be used to test existing systems and techniques and to drive new developments. There is, however, a perennial danger in such an approach in that the focus on these problems becomes an end in itself, and the connection with the reality of which the problem is but one example becomes increasingly tenuous. It is therefore necessary from time to time to relate theoretical treatments to the real problems of the domain.

AI and Law is not exempt from such dangers. Over time a distinctive approach to modelling legal reasoning has developed. This strand of work has its origins in the HYPO system of Rissland and Ashley [2], and its developments through the CATO system of Ashley and Aleven [1], and the current work of Ashley and Brüninghaus [3]. This work, firmly grounded in case-based reasoning, has also inspired some of those who come from a rule based and logic background: including Prakken, Sartor, Hage and Bench-Capon. In recent years the approach has begun to include notions of theory construction (originally advocated by McCarty, e.g. [17]), purpose (originally advocated by Berman and Hafner [9]) and dialogue (introduced by Gordon [14]). The approach typically includes the following features:

- Cases are represented as collections of factors. These are not facts, but rather patterns of facts which have some legal significance;
- Legal reasoning is essentially argumentation: on the basis of the factors present in the case, the plaintiff and defendant put forward arguments as to why the case should be decided in their favour;
- These arguments are often presented as a dialectical exchange between the parties;
- A legal decision involves evaluating competing arguments to see which should be accepted;
- Strengths of arguments are based on the purposes served by accepting the argument concerned.

This approach has produced much interesting and sophisticated work, but some doubts remain as to the relevance of this conception to legal reasoning as it is actually reported. These doubts were forcefully raised in a paper by Branting [10] and also formed part of his keynote address at ICAIL 2005. He identified six points which suggested that theory was out of alignment with practice. This is a challenge to which those of us who are working within the approach outlined above need to respond, and this paper is an attempt to do so.

We do so in the context of our recent work on persuasive argumentation in law. We view legal reasoning as taking place on a number of different but interconnected levels, with each level having its own important role to play in the process. In earlier work [6] we have demonstrated this by showing how a general account of practical reasoning can be applied to arguing about legal cases. In that paper we demonstrated our approach by providing a computational reconstruction, using BDI agents, of the reasoning of the majority and dissenting opinions for *Pierson vs Post*, a particular case from property law. The reconstruction of this case suggested that the reasoning involved can be separated into three distinct levels: factual and normative levels and a level connecting the two, with conclusions at one level forming premises at the next, as we discussed in detail [4,6]. In this paper we discuss how this framework can be used to address Branting's challenges, and show how it fits with other accounts from AI and Law to provide some important distinctions in the process of legal reasoning.

In section 2 we discuss Branting's paper [10] and the six points of conflict he identified between the views of the AI and Law research community and the actual practice of law in precedent-based reasoning. In section 3 we give a brief overview of our account of the levels of reasoning involved in legal cases and we discuss how they can help to explain some of the problems identified by Branting. In section 4 we discuss how the observations made in the previous sections fit with the construction and use of theories, as described by Bench-Capon and Sartor in [8], and relate this to some other leading AI and Law systems. Finally, we offer some concluding remarks in section 5.

2. Empirical Investigations Regarding Precedents

In [10] Branting argues that formal and computational models of legal systems do not closely match real world practices. In particular, he maintains that this problem is most prevalent in computational models of precedent-based legal reasoning, as these models do not correspond to the problem-solving practices that exist in the Anglo-American legal system. In order to address this problem, Branting proposes that the AI and Law re-

search community should shift its focus towards task analysis and increased empirical investigation to ensure that computational models of legal practices do in fact reflect real world legal reasoning. In response to this view, we aim to show how our previous investigations into the different levels of reasoning that occur in legal cases can in fact provide a realistic model for computational representation of case-based reasoning and explain the phenomena observed by Branting. Before we discuss how our proposals reflect this view, we first examine in more detail Branting's observations regarding the inconsistencies that exist between research and actual practices involving precedent-based legal reasoning.

2.1. Precedent-Based Reasoning in Practice

In [10] Branting identifies six points of contention that demonstrate how the conduct of practices in courts actually conflicts with widely accepted views of precedent-based reasoning from the research community. We summarise Branting's six observations below.

1. **The outcome of disputes on questions of law is highly predictable.**

   Whereas the current AI and Law approach sees almost all cases as debatable with arguments on both sides, in practice appeals that are brought before courts that operate towards the lower end of the appellate system (which is where most appeals are dealt with) are likely to have predictable outcomes. The predictability of outcomes does decrease the higher the court of jurisdiction, though only a very small number of cases are actually considered at the highest level. Cases considered at the highest level (e.g. the US Supreme Court) are nearly always unique and highly irregular, making their outcome more unpredictable.

2. **Uncertainty arises primarily from disagreements about the characterisation of facts, not from ambiguity about the precedential effect of cases.**

   Whereas the current AI and Law approach takes the factors present as given, and concentrates on the applicability and use of precedents, the role that precedent cases play in appeal decisions is typically clear to Judges. Any uncertainty involved in the decision is primarily manifest through disagreements about the facts of the case and how they should be described, not through the role that the precedents play.

3. **Judicial decisions are almost always justified by precedent, not by statutory rules plus principles of construction.**

   Whereas the current AI and Law approach emphasises interpretation, virtually all appellate decisions are made solely through simple reference to precedents. In cases where there are no precedents then the court may discuss treatises by noted authorities on the matter.

4. **Judicial decisions are not frozen dialectic.**

   Whereas the current AI and Law approach emphasies argument and debate, it is the role of Judges to make decisions about cases that are based upon their inter-

pretation of the law, rather than upon the particular arguments presented to them by attorneys. Decisions are based upon the relevant laws and precedents, not the oral arguments presented.

5. **Cases are not decided upon the basis of teleological factors.**

   Whereas some current work in AI and Law looks to justify priorities between arguments through considerations of purpose, judicial decisions rarely consider the underlying social goals and values that are relevant to the decision. Reported decisions are solely based upon consideration of factual evidence and they take no account of teleological factors.

6. **Factor-based argumentation occurs in only a small minority of cases.**

   Whereas the current AI and Law approach relies on cases being represented as sets of factors, this is by no means universal. In certain areas of law where there are no well-defined rules (such as tax law) the courts can indeed identify a collection of factors that are relevant to the case. This type of argumentation however, actually occurs in only a small proportion of cases.

Given these observations that seem to reveal inconsistencies between legal reality and the focus of AI and Law research, we believe that we need to shed some light on the reasons as to why these conflicts exist. We do so by turning to our previous work which views legal reasoning as being divided into three levels. In the next section we will briefly recapitulate this work and show how the division of legal reasoning in this manner can account for some of the issues raised by Branting.

3. Levels of Reasoning in Legal Cases

In recent work [6] we have explored the representation of legal cases as practical reasoning. We have taken a general account of practical reasoning developed by the authors [5] and applied it to arguing about legal cases. As a concrete example, we provided a reconstruction of the reasoning of the majority and dissenting opinions for a particularly famous legal case in property law: *Pierson vs Post*, 3 Cai R 175 2 Am Dec 264 (Supreme Court of New York, 1805). This reconstruction successfully made use of Belief-Desire-Intention (BDI) agents to replicate the contrasting views involved in the actual decision. An interesting point that emerged from this application was that it showed how the reasoning involved can be separated into three distinct levels: a factual level, a normative level, and a level connecting the two, with conclusions at one level forming premises at the next. In [4] we attempted to generalise these findings by examining the nature and significance of these levels of reasoning in more detail and we provided further examples of their occurrence in law. Here we will give a brief recapitulation of these levels before discussing how they relate to Branting's observations.

3.1. Levels of Reasoning

In the BDI agent reconstruction of *Pierson vs Post* described in [6], the reasoning involved naturally separated into the following three levels:

- **Level 1:** the level of facts about the world, at which desires are derived, which concerns how the law should be.
- **Level 2:** the level at which the legal system connects with the world to achieve these desires, by the appropriate attribution of intermediate concepts.
- **Level 3:** the level of pure legal concepts at which a settled theory is applied.

These levels are connected in that conclusions at lower levels serve as premises at higher levels. Level 3 is concerned solely with legal concepts and the rights they confer. Level 2 is concerned with the ascription of these legal concepts, given the particular facts of the case under consideration. Here arguments for and against ascription of the legal concepts can come either from precedents or from purposes derived from reasoning in the bottom layer (Level 1). At Level 1 people reason about the world in order to determine what the law should be, and conclusions from this level are used at Level 2. These levels and their relationship are shown diagrammatically in Figure 1 below:

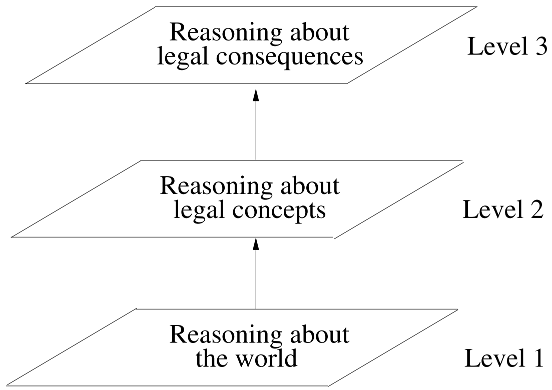

Figure 1. The three levels of legal reasoning emerging from the reconstruction of *Pierson vs Post*.

Related work has also observed these three levels, as we discussed in detail in [4]. In that paper we examined how our levels relate to Lindahl's work [15,16], which makes a distinction between fact situations, intermediate legal concepts, and the consequences that flow from them. We also discussed the work of Breuker and den Haan [12], which explored the need to represent knowledge of the world, knowledge of legal concepts and the connections between them. Additionally we examined Ashley and Brüninghaus' work [3] which stressed the vital role played by intermediate concepts in predicting the

outcome of cases. The idea there is that the decision is predicted on the basis of the intermediate concepts which apply, and the factor based reasoning taken from CATO is mainly used to establish them. This has clear correspondences with our Level 3, where the intermediate concepts are applied, and our Level 2 where the presence of intermediate concepts is established. However, their approach uses case-based reasoning whereas we rely on a model of practical argumentation, and they also make no appeal to purpose. What we add is the ability to motivate the ascription of intermediate concepts through desirable purposes, determined at Level 1.

3.2. Relation to Branting's Observations

We now return to Branting's observations regarding the conflicting views of precedent-based reasoning between the research community and the actual practice of law. Here we will describe how each of the points of contention raised by Branting can be explained in terms of the particular levels of reasoning, and the issues involved at each of these levels, that we have identified and described above. A key point, as discussed in [4], is that the decision is very often stated in therms of the reasoning at Level 3, occasionally descending to Level 2 and very rarely (and then typically only at the highest level of appeal) reaching Level 1.

1. **The outcome of disputes on questions of law is highly predictable.**

   In our model we can account for this predictability by looking at the reasoning in Level 3. At this level the only reasoning involved concerns the legal decision to be taken, given the facts from the previous levels. Level 3 solely concerns the establishment of a legal conclusion of the case and the issues at this level do not involve disputes over facts. At this stage of the reasoning the facts of the case, including any relevant precedents, have already been established and the decision to be taken should be straightforward, given the relevant applicable laws on the matter. So, we would expect that the outcome of a case would be highly predictable at this level. In *Pierson vs Post*, for example, the majority decision does little more than state that since Post had no possession he has no remedy, which makes the outcome look uncontestable.

2. **Uncertainty arises primarily from disagreements about the characterisation of facts, not from ambiguity about the precedential effect of cases.**

   This issue is characterised through the transition from Level 1 to Level 2 reasoning in our model. Here, it must be decided whether the facts of the case in question can instantiate a particular legal concept. The arguments involved are concerned with determining which intermediate concepts apply and also any precedents that are relevant. It is at this stage that the uncertainty of how the facts are to be described must be resolved before being able to proceed to Level 3 reasoning, which solely involves the legal, not factual concepts. Again taking *Pierson vs Post* as our example, the uncertainty on the case revolves around whether Post should be deemed to possess the fox: the facts are not in dispute, only how they

should be characterised in terms of intermediate concepts.

3. **Judicial decisions are almost always justified by precedent, not by statutory rules plus principles of construction.**

   This point relates to Levels 2 and 3 from our model. Here, there are no discussions regarding the purposes and values which motivate the legal decision. Such purposes are considered in Level 1 reasoning, which concerns how the law should be. As the values are fixed once Levels 2 and 3 are reached, the legal decision need make no reference to them, other than to cite the precedent which fixed them.

4. **Judicial decisions are not frozen dialectic.**

   Again, as for the first point, the issue raised here involves the distinction between Level 2 and Level 3 reasoning. As discussed earlier, the reasoning at Level 3 solely involves making legal decisions, given the facts of the case from Level 2, and a settled theory with respect to the precedents. Level 2, where the dialectical exchange of arguments takes place, resolves the questions of interpretation, and only the conclusions are required to state the decision at Level 3.

5. **Cases are not decided upon the basis of teleological factors.**

   As noted in point 3 above, decisions about legal cases do not involve discussion of purposes, social principles and values. In our model, this lack of concern with purpose is reflected by its absence from the reasoning of Levels 2 and 3. However, if we descend to Level 1 reasoning, then this does involve discussion of teleological factors. At this level, facts about the world are discussed and arguments are presented to determine what the law *should* be, if it is to fulfil its purposes. Such discussions typically do not take place in a courtroom (though in the case of *Pierson vs. Post* Livingston (who gave the minority opinion), who needs to extend the existing law to win for his side, does argue a point for what he believes the law should be in terms of purpose). Normally however, such considerations are viewed as a matter to be determined by the legislative body.

6. **Factor-based argumentation occurs in only a small minority of cases.**

   In cases where factor-based argumentation occurs, resolution of disagreements will generally be brought about by descending to Level 1 reasoning. Here, the individual factors are considered and preferences will be formed between the factors, with respect to their individual applicability for the law. As this type of reasoning generally only occurs in areas where there are no well-defined rules, then it is no surprise that the reasoning needs to descend to Level 1. And as Level 1 reasoning is normally taken as settled when a decision is given, then it is also not surprising that such factor-based argumentation only occurs in a small minority of reported decisions. Essentially the factor based considerations are subsumed in the theory being applied, and can be expressed by citing the landmark precedents

in which the debate occurred, without redoing the teleological reasoning which gave rise to the decision.

We believe that the issues raised by Branting can be accounted for in terms of our model identifying the different levels involved in legal reasoning, and which level of reasoning is being reported in the given context. In recognising how and why there are disparities between what we find in AI and Law research and what appears in decisions on cases, we can begin to address the issues in current theories and models of AI and Law.

We now leave our discussion of Branting's work and we turn our attention to the topic of theory construction. In the next section we will show how the levels of reasoning described in section 3.1 are also reflected in the process of theory construction, and also comment on the relation to landmark work in AI and Law, including the HYPO [2] and CATO [1] systems.

4. Relation to Theory Construction

In [8] Bench-Capon and Sartor provide a detailed description of the process involved in constructing theories making use of factors and values. In their model they make use of factors, each associated with some value, which provide a way of describing cases, and these factors are seen as grounding defeasible rules. Preferences between factors are expressed in past decisions, which indicate priorities between the rules. From these priorities it is then possible to abduce certain preferences between the values involved. This model of case law can thus be seen as a process which reveals a preference ordering over values. Bench-Capon and Sartor depicted this process diagrammatically, as shown below in Figure 2. We have added one more box to the original diagram which represents the process of domain analysis required to identify factors and describe cases in terms of them.

As described by Bench-Capon and Sartor, the process of theory construction begins through examination of precedent cases. The next levels are then constructed by identifying the rule-preferences that are revealed in the precedent cases, and the value preferences which these rule preferences show. The theory has now been constructed and it can be used to explain the precedents involved, as well as enabling predictions to be given for new cases.

The levels of legal reasoning we described in section 3.1 fit nicely with Bench-Capon and Sartor's account of theory construction and use in factor-based domains, as we will now elaborate.

From Figure 2, we can view the process of factor-based reasoning as an interaction taking place between the different components of the model. At the start, in the bottom left hand corner, Level 1 reasoning is conducted for the domain analysis part of the process. Here, decisions are made to determine which factors are to be included in the theory, which values are served by including them, and which factors are present in the various cases. The relevant factors will be determined by deciding which values are to be promoted and consequently, which factors promote these values. These factors form the input to the next stage, where Level 2 reasoning occurs (the box labelled "Outcomes and factor descriptions in decided cases"). At this point, reasoning is conducted to establish

the relative importance and role that each factor played in the decision of the precedent case and this produces a preference ordering on the factors and their combinations. The outcome of this reasoning is represented by the "reveal" relation which takes us from the position where the factors of the precedent cases are considered, to the position where we have a preference ordering between sets of factors (the middle box in the diagram). This preference ordering between the sets of factors will now enable us to establish a value preference ordering, characterised by Level 1 reasoning and shown at the top of the diagram. Recall from the discussions in the previous section that Level 1 is the only level of our model that involves consideration of values. So, the reasoning here reveals the values that are promoted by the factors, though this level of reasoning is generally not of primary concern in the decision making stage of the legal reasoning. Having established a factor and value preference ordering, we now have a complete theory and this can be used to determine outcomes for new cases. This is where the Level 3 reasoning commences and it is at a purely legal level. Here we move down the "determine" relation (on the right hand side of the diagram) where the new legal decisions are made, based upon the constructed theories. If the law is seen as settled, the theory can be taken as given, and the decision in a given case will be couched in terms of the application of this established theory, and so need contain no reference to values, or to the balancing of factors which led to the ascription of the intermediate concepts.

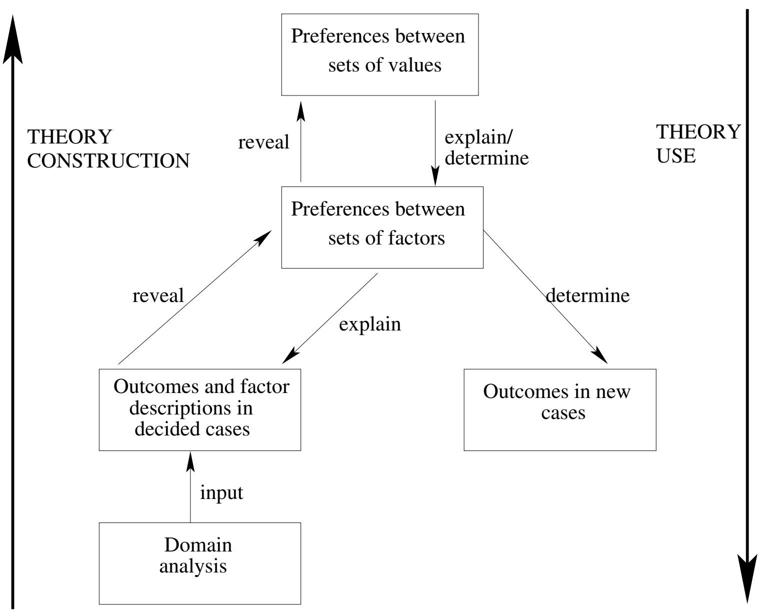

Figure 2. Construction and Use of Theories, adapted from [8].

We can now look at some established systems in AI and Law and see how they address various parts of the process. A simple rule based system, such as [20] is concerned only with the transition from a settled theory to the outcome of cases. Such systems can thus be seen as embodying Level 3 reasoning only, and produce the kind of decisions which confine themselves to the concerns raised by Branting.

A case based system such as CATO involves the transition from precedents to preferences, building arguments as to which intermediate factors (called "abstract factors" in

CATO) apply. Ashley's more recent work with Brüninghaus, the IBP system [3], extends the scope to the application of these preferences, so as also to predict case outcomes, by following the "determines" link to "outcomes in new cases". CATO goes beyond what is seen in routine decisions, because it is targeted at law students, and needs to address the Level 2 reasoning which explains the ascription of intermediate concepts (represented by CATO's abstract factors). Argumentation is crucial at this point because the preferences revealed must be tested and justified in terms of their ability to explain and cover the remaining precedents. Prakken and Sartor's reconstruction of HYPO [19] covers a similar area.

The Level 1 reasoning representing the transition from rule preferences to value preferences introduces teleological notions and was first advocated by Berman and Hafner [9]. These notions are also incorporated in work by Prakken [18] and Bench-Capon [7]. Again the preferences revealed can be disputed and must be justified, accounting for the need for argumentation in this transition. Here, however, we are dealing with the underpinnings of the theory, more the concern of commentaries than judgements, except where we are dealing with a case where the law needs to be extended or changed. Such cases are typically only encountered at the highest levels of appeal. Dialectics are ways of conducting arguments, and so they too will only appear at Levels 1 and 2.

Viewed in this way, we can see the AI and Law approach as attempting to describe legal reasoning in terms of the process required to "reason from first principles", as opposed to the application of tried and trusted rules. We can see a similar dichotomy in AI approaches to science: where we can either attempt qualitative reasoning on a "deep model", or use heuristics summarising the behaviour of such a model. In this field attempts have been made to derive the heuristics from the deep model [11]. In AI and Law this can be seen as the underlying approach of [13], where Level 3 and Level 2 reasoning go to construct a theory which is used to produce a Prolog program which is then executed to perform the Level 3 reasoning. As Branting's first observation indicates, such "first principles" reasoning is deployed in some cases (albeit rare and in the higher courts), so it is legitimate for AI to model this reasoning, even if the fruits of such reasoning can be delivered in less sophisticated systems to handle routine cases.

One transition not addressed in any of these systems is the allocation of factors to cases, represented by the box labelled "domain analysis" in Figure 2. This is germane to Branting's second criticism, that uncertainty often arise from disputes about the factual characterisation of cases. CATO-type systems do address the attribution of intermediate concepts on the basis of factors, but the factors are taken as given. This may well point to a significant gap in the current theoretical work, since it is the mapping from "raw" facts to these stylised fact patterns that is a key part of the lawyers skill. The use of dimensions in HYPO, allows some limited consideration of this problem, in that its dimensions allow for some debate as to whether a factor is present (e.g. posing the question: just what is required before we accept that the plaintiff took reasonable measures to protect his "Trade Secret?"). None the less we are still a long way from techniques allowing us to move from the kind of narrative that a client might present, to the kind of fact description that a lawyer can derive from such narratives, as described in detail in [10].

5. Concluding Remarks

In this paper we have used our previous work which has examined the different levels involved in reasoning about legal cases [4]. The presence of these different levels emerged from a BDI agent reconstruction of the famous property law case of *Pierson vs Post*. However, these levels have also proved to be instructive when considering their relation to other research from the AI and Law community. We examined some observations made by Branting in [10] which reveal apparent inconsistencies between legal practice and computational models and theories of AI and Law. We believe that our model can reveal where and why these inconsistencies arise and thus identify important distinctions that need to be made when modelling legal reasoning. We have also used this account to explain some differences in scope in some of the existing leading systems in AI and Law.

Acknowledgements

Katie Atkinson is grateful for support from the EPSRC. Trevor Bench-Capon is grateful for partial support from the European Commission, through project APSIC (IST-FP6-002307).

References

[1] V. Aleven. *Teaching Case Based Argumentation Through an Example and Models*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1997.

[2] K. D. Ashley. *Modeling Legal Argument*. MIT Press, Cambridge, MA, USA, 1990.

[3] K. D. Ashley and S. Brüninghaus. A predictive role for intermediate legal concepts. In D. Bourcier, editor, *Proceedings of the Sixteenthth Annual Conference on Legal Knowledge and Information Systems (Jurix 2003)*, pages 153–162, Amsterdam, The Netherlands, 2003. IOS Press.

[4] K. Atkinson and T. Bench-Capon. Levels of reasoning with legal cases. In P. E. Dunne and T. Bench-Capon, editors, *ICAIL 2005 Workshop on Argumentation in AI and Law*, IAAIL Workshop Series, pages 1–11, Nijmegen, The Netherlands, 2005. Wolf Legal Publishers.

[5] K. Atkinson, T. Bench-Capon, and P. McBurney. Justifying practical reasoning. In F. Grasso, C. Reed, and G. Carenini, editors, *Proceedings of the Fourth International Workshop on Computational Models of Natural Argument (CMNA 2004)*, pages 87–90, Valencia, Spain, 2004.

[6] K. Atkinson, T. Bench-Capon, and P. McBurney. Arguing about cases as practical reasoning. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL 2005)*, pages 35–44, New York, NY, USA, 2005. ACM Press.

[7] T. Bench-Capon. Try to see it my way: Modelling persuasion in legal discourse. *Artificial Intelligence and Law*, 11 (4):271–87, 2003.

[8] T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150 1-2:97–143, 2003.

[9] D. H. Berman and C. D. Hafner. Representing teleological structure in case-based legal reasoning: the missing link. In *Proceedings of the Fourth International Conference on AI and Law (ICAIL 1993)*, pages 50–59, New York, NY, USA, 1993. ACM Press.

[10] K. L. Branting. An agenda for empirical research in AI and law. In *Working Papers of the ICAIL 2003 Workshop on Evaluation of Legal Reasoning and Problem-Solving Systems*, pages 28–35, Edinburgh, UK, 2003.

[11] I. Bratko, I. Mozetič, and N. Lavrač. *KARDIO: a study in deep and qualitative knowledge for expert systems*. MIT Press, Cambridge, MA, USA, 1989.

[12] J. Breuker and N. den Haan. Separating world and regulation knowledge, where is the logic? In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL 1991)*, pages 92–97. ACM Press: New York, USA, 1991.

[13] A. Chorley and T. Bench-Capon. AGATHA: Automated construction of case law theories through heuristic search. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL 2005)*, pages 45–54, New York, NY, USA, 2005. ACM Press.

[14] T. F. Gordon. The Pleadings Game: An exercise in computational dialectics. *Artificial Intelligence and Law*, 2:239–292, 1994.

[15] L. Lindahl. Deduction and justifi cation in the law. The role of legal terms and concepts. *Ratio Juris*, 17 2:182–202, 2004.

[16] L. Lindahl and J. Odelstad. Normative positions within an algebraic approach to normative systems. *Journal of Applied Logic*, 17 2(1):63–91, 2004.

[17] L. T. McCarty. An implementation of Eisner v Macomber. In *Proceedings of the Fifth International Conference on AI and Law (ICAIL 1995)*, pages 276–286, New York, NY, USA, 1995. ACM Press.

[18] H. Prakken. An exercise in formalising teleological case-based reasoning. *Artificial Intelligence and Law*, 10 (1-3):113–133, 2002.

[19] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6 (2-4):231–287, 1998.

[20] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Norm Modifications in Defeasible Logic

Guido Governatori [a,1], Monica Palmirani [b], Regis Riveret [b], Antonio Rotolo [b] and Giovanni Sartor [b]
[a] *School of ITEE, University of Queensland*
[b] *CIRSFID, University of Bologna*

Abstract. This paper proposes a framework based on Defeasible Logic (DL) to reason about normative modifications. We show how to express them in DL and how the logic deals with conflicts between temporalised normative modifications. Some comments will be given with regard to the phenomenon of retroactivity.

Keywords. Norm Modifications, Defeasible Logic

1. Introduction

This paper proposes a logical framework based on DL to deal with norm-modifications. We have different types of modifications according to how they affect the law. The impact might concern, e.g., the law text, its scope, or the time of its force, efficacy, or applicability [3]. So we can identify four main categories of modifications: (1) textual changes, (2) change of the norm scope, (3) temporal changes, (4) normative system changes. Textual changes intervene when a law is repealed, replaced, integrated or relocated. Changes of scope might be consequent on derogation or extension. Temporal changes impact on the date in force, date of efficacy or date of application of the destination norm. As a first step, in this paper we will focus on three kinds of modifications: substitution (which replaces some textual components of a provision with other textual components, or a provision with another provision), derogation (the derogating provision limits the effects of the derogated provision), and annulment (which cancels *ex tunc* a provision and prevents it to produce any normative effect).

In particular, we are interested in logically investigating the following issues: (A) Conditional modifications, which apply under, and are conditioned to, the occurrence of some uncertain events. (B) The notion of conflict between textual modifications and the logical strategies for solving them. (C) The concept of time-forking of modifications; indeed, there are modifications affecting the effects previously obtained by other modifications. This is the case, in particular, when we have retroactive modifications. Retroactive modifications lead to the forking (branching) of the versioning chain of modifications in order to keep trace both of the past modifications and of the new versioning chains (see [2]). The system is based on the following building blocks or assumptions, which are needed in DL to correctly represent the dynamics of normative systems. **Normative Conditionals:** It is possible to distinguish between different kinds of normative conditional. Here we will identify the following types of normative rules [1]: (1) *Rules for persistent*

[1]Correspondence to: Guido Governatori, School of ITEE, The University of Queensland, Brisbane, Australia, QLD 4072, guido@itee.uq.edu.au.

obligations, which, if applicable, permit to infer literals to be modalised by obligations that persist unless some other, subsequent, and incompatible events or states of affairs terminate them. For example, the obligation of paying the damages in a car crash will hold until such damages have been paid. (2) *Rules for co-occurrent obligations*, which allow for the inference of obligations which hold on the condition and only while the antecedents of these rules hold. For example, the obligation not speak loud in the church will hold only when the agent is in the church. (3) *Rules for counts-as links*, which express the idea of institutional power. For example, if i signs a document on behalf of her boss j, such a document is as it were signed by j only if i has been empowered to do this: i's signature counts as j's signature [1]. Here we will simply view the counts-as link as a normative conditional whose consequences are not necessarily deontic. In this context, modifying meta-norms, namely rules regulating norm-modifi cations, will be represented as counts-as rules. **Time:** Norm application and modifi cation take place along the axis of time. Since we will operate in a temporalised setting, we will not only impose that obligations be temporalised, but also that rules and any literal are labelled by time instants. In particular, a rule is represented as $(a:t \Rightarrow b:t'):t''$. t and t' indicate the time at which a and b hold, while t'' is the time of the rule being in force. We will assume here the time to be linear and discrete. **Normative provisions:** In general, complex normative provisions have an internal structure and can be decomposed into a number of nested units. Here we will assume that the rules of our logical theory correspond to the atomic normative provisions constituting complex provisions. This simplifi cation will keep the system manageable.

2. Basic Formal Language

Defeasible logic is a flexible sceptical non-monotonic formalism that has proven able to represent various aspects of normative reasoning. We adapt here the machinery developed in [1] to represent temporalised normative positions to reason both on the normative provisions in a legal systems and the meta-norms describing the modifi cations of legal texts.

Our language is based on a (numerable) set of atomic proposition $Prop = \{p, q, \dots\}$, a set of rule labels $\{r_1, r_2, \dots\}$, a discrete totally ordered set of instants of time $\mathscr{T} = \{t_1, t_2, \dots\}$, the modal operator ***Obl*** of obligation, and the negation sign $\neg$. A plain literal is either an atomic proposition or the negation of it. If l is a plain literal then ***Obl****l* and $\neg$***Obl****l* are modal literals. A literal is either a plain literal or a modal literal. Given a literal l with $\sim l$ we denote the complement of l, that is, if l is a positive literal p then $\sim l = \neg p$, and if $l = \neg p$ then $\sim l = p$. Finally we introduce the notion of temporal literals. A temporal literal is a pair $l:t$ where l is a literal and t is an instant of time. Intuitively the meaning of a temporal literal $l:t$ is that l holds at time t. Knowledge in defeasible logic can be represented in two ways: facts and rules. *Facts* are indisputable statements, represented either in form of states of affairs (literal and modal literal) and actions that have been performed. For example, "John is a minor". In the logic, this might be expressed as $Minor(John)$. A *rule* is a relation between a set of of premises (conditions of applicability of the rule) and a conclusion. In this paper the admissible conclusions are either normative provisions (obligations, permissions) or rules themselves, in addition the conclusions and the premises will be qualifi ed with the time when they hold. We consider two classes of rules: counts-as rules and deontic rules. Counts-as rules describe the inference mechanism of the institution on which norms are formalised and can be

used to establish definitions as well as conditions for the creation and modification of other rules. Deontic rules on the other hand give the conditions under which normative qualifications (obligations and permissions) hold.

Beside the above classification rules can be partitioned according to their strength into *strict rules* (denoted by $\rightarrow$), *defeasible rules* (denoted by $\Rightarrow$) and defeaters (denoted by $\rightsquigarrow$). Strict rules are rules in the classical sense: they are monotonic and whenever the premises are indisputable so is the conclusion. Defeasible rules, on the other hand, are non-monotonic: they can be defeated by contrary evidence. Finally defeaters are the weakest rules: they do not support conclusions, but can be used to block the derivation of opposite conclusions. Thus we define the set of rule Rules using the following recursive definition:

- A rule is either a counts-as rule or a deontic rules or the empty rule $\perp$, where
- If r is a rule and $t \in \mathscr{T}$, then $r : t$ is a temporalised rule. (The meaning of a temporalised rule is that the rule is in force at time t).
- Let A be a finite set of temporal literals, C be a temporal literal and r a temporalised rule, then $A \hookrightarrow C$, $A \hookrightarrow r$ and $A \hookrightarrow \sim r$ are counts-as rules (henceforth we use $\hookrightarrow$ as a metavariable for either $\rightarrow$ when the rule is a strict rule, $\Rightarrow$ when the rule is a defeasible rule, and $\rightsquigarrow$ when the rule is a defeater).
- Let A be a finite set of temporal literals and C be a temporal plain literal. Then $A \hookrightarrow_O C$ is a deontic rule (henceforth we use $\hookrightarrow_O$ as a metavariable for either $\rightarrow_O$ when the rule is a strict rule, $\Rightarrow_O$ when the rule is a defeasible rule, and $\rightsquigarrow_O$ when the rule is a defeater).

For a rule r we will use $A(r)$ to indicate the body or antecedent of the rule and $C(r)$ for the head or consequent of the rule. The above inductive definition makes it possible to have nested rules, i.e., rules occurring inside other rules. However, it is not possible for a rule to occur inside itself. Thus for example, the following is a rule

$$p : t_p, \boldsymbol{Obl}q : t_q \Rightarrow (\boldsymbol{Obl}p : t_p \Rightarrow_O s : t_s) : t_r \tag{1}$$

Eq. (1) means that if p is true at time t_p and q is obligatory at time t_q, then the deontic rule $\boldsymbol{Obl}p : t_p \Rightarrow_O s : t_s$ is in force at time t_r. The intuition we want to represent is that deontic rules are meant to introduce obligations. We do not admit deontic literals and rules as conclusions of deontic rules since the meaning of modalised rules and nested deontic modalities is not clear. Every temporalised rule is identified by it rule label and its time. Formally we can express this relationship by establishing that every rule label r is a function $r : \mathscr{T} \mapsto$ Rules.

Thus a temporalised rule $r : t$ returns the value/content of the rule 'r' at time t. This construction allows us to uniquely identify rules by their labels, and to replace rules by their labels when rules occur inside other rules. In addition there is no risk that a rule includes its labels in itself. For example if we associate the temporal rule $(\boldsymbol{Obl}p : t_p \Rightarrow_O s : t_s) : t_r$ to the pair $r_1 : t_r$ then we can concisely rewrite eq. (1) as

$$p : t_p, \boldsymbol{Obl}q : t_q \Rightarrow r_1 : t_r \tag{2}$$

We have to consider two temporals dimension for norms in a normative systems. The first dimension is when the norm is effective in the normative system, and the second

when the norm is in force in the normative system. So far temporalised rule capture only one dimension, the effectiveness one. To cover the other dimension we introduce the notion of temporalised rule with viewpoint. A temporalised rule with viewpoint is an expression $s@t$ where s is a temporalised rule, and $t \in \mathscr{T}$. Thus the expression $r_1 : t_1@t_2$ represents a rule r_1 in force at time t_2 and effective from time t_1.

An issue we need to consider here is that we have two different types of normative conditionals: conditionals that initiate an action or a state of affairs which persists until an interrupting event occurs, and conditionals where the conclusion is co-occurrent with the premises. To represent this distinction we introduce a further distinction of rules, orthogonal to the previous one, where rules are partitioned in persistent and transient rules. A persistent rule is a rule whose conclusion holds at all instants of time after the conclusion has been derived, unless interrupting events occur; transient rules, on the other hand, establish the conclusion only for a specifi c instant of time. We use the following notation to differentiate the various types of rules: with $\hookrightarrow^t_O$ we represent a transient deontic rule, $\hookrightarrow^p_O$ a persistent deontic rule, $\hookrightarrow^t$ a transient counts-as rule, and $\hookrightarrow^p$ a persistent counts-as rule.

Given a set R of rules, we denote the set of strict rules in R by R_s, the set of strict and defeasible rules in R by R_{sd}, the set of defeasible rules in R by R_d, and the set of defeaters in R by R_{dft}. $R[q:t]$ denotes the set of rules in R with consequent $q:t$. We will use R^c for the set of counts-as rules, R^O to denote the set of deontic rules. The set of transient rules is denoted by R^t and the set of persistent rules by R^p. Finally we assume a set of rule modifi ers. A rule modifi er is a function $m : \mathscr{T} \times \text{Rules} \mapsto \mathscr{T} \times \text{Rules}$.

The combination of the above two constructions allows us to use rule modifi ers on rule labels. Thus $m(r_1 : t_1) : t_2$ returns the rule obtained from r_1 as such as time t_1 after the application of the modifi cation corresponding to the function m and the results refers to the content of the rule at time t_2. Given this basic notion of rule modifi er, we can defi ne some specifi c norm-modifi cations: Annulment, PSubstitution and TSubstitution (partial and total substitution), and Derogate (derogation). Let see their intuitive reading. Recall that $A(r)$ denotes the set of literals occurring in the antecedent of a rule r and $C(r)$ denotes the consequent of r. Suppose $r \in R$ is a generic defeasible rule (either counts-as or deontic) $(a_1 : y_1, \dots a_n : y_n \Rightarrow b : j) : t$. Then $\text{Annulment}(mr, r) : t'$ says that r is annulled at t' by the meta-rule mr.

Let $\mathscr{A}_r$ and $\mathscr{C}_r$ specify the substitutions of literals to be applied, within the rule r, respectively in its antecedent and in its consequent. Where $l \geq 1$ and $l + m \leq n$,

$$\text{PSubstitution}(mr, r, \mathscr{A}_r(a'_l : x_l / a_l : y_l, \dots, a'_{l+m} : x_{l+m} / a_{l+m} : y_{l+m})) : t'$$
$$\text{PSubstitution}(mr, r, \mathscr{C}_r(b' : k / b : j)) : t'$$

say, respectively, that we operate, at t' through mr, a substitution which replaces a subset or all literals in the antecedent of r with other literals $a'_l : x_l, \dots, a'_{l+m} : x_{l+m}$, and $b : j$ with $b' : k$ in the consequent of r. The new version of r will hold at t'.

$$\text{TSubstitution}(mr, r'/r, A(r') = \{d_1 : s_1, \dots, d_o : s_o\}, C(r') = e : z) : t'$$

indicates total substitution of r, i.e. that r is replaced, at t' through mr, by a different rule r' holding at t' and having the antecedent and consequent as specifi ed within the predicate[1].

[1]To simplify the discussion, we will assume that r' will have the same strength of r.

Let $\mathscr{D}$ and $\mathscr{E}$ indicate the additional literals used, respectively in the antecedent of r' and r'', to specify how to derogate to r. Then

$$\text{Derogate}(mr, r, r', r'', \mathscr{D}(q_1 : z_1, \ldots, q_m : z_m), \mathscr{E}(b' : k)) : t'$$

indicates derogation to r. Derogation may affects "spatial" condition of application of a norm r or its conceptual range of application. In the first case, a norm r holding at the national level may be derogated by a norm r' in the event we operate, for example, within a regional context. In the second case, derogation may affect the conceptual range of r. In the first sense, for example, if r states that the tax rate in Italy corresponds to 30% of total income, a rule r' will state that, if one is resident in Emilia Romagna, under the same conditions, the tax rate will be 25% of the total income. In both cases, anyway, we have to identify logical exceptions of r. Thus the predicate says that r is derogated, at t' through mr, by a rule r' holding at t', which last includes in its antecedent the same conditions of r plus some additional conditions $q_1 : z_1, \ldots, q_m : z_m$. Also, such a rule r' produces the effect b' rather than b. However, under r's conditions and $q_1 : z_1, \ldots, q_m : z_m$ we should also block the derivation of the consequent of r: this is done by rule r'', a defeater, which holds as well at t' and that will have in the consequent the negation of the consequent of r^2. Let us now characterise the modifiers that correspond to the above predicates. If $r : (a_1 : y_1, \ldots a_n : y_n \Rightarrow b : j) : t$

- the modifier corresponding to Annulment$(mr, r) : t'$ assigns the empty rule $r : (\perp) : t'$ to r as holding at t. The rule r is thus dropped at t' from the system and so, at t', r is not in force.
- the modifier corresponding to

  $$\text{PSubstitution}(mr, r, A(a'_l : x_l / a_l : y_l, \ldots, a'_{l+m} : x_{l+m} / a_{l+m} : y_{l+m})) : t'$$

  assigns, to r at t, the following rule

  $$r : ((A(r) - \{a_l : y_l, \ldots, a_{l+m} : y_{l+m}\}) \cup \{a'_l : x_l, \ldots, a'_{l+m} : x_{l+m}\} \Rightarrow b : j) : t'$$

  while the modifier corresponding to PSubstitution$(mr, r, \mathscr{C}(b' : k / b : j)) : t'$ assigns, to r at t, the rule $r : (a_1 : y_1, \ldots a_n : y_n \Rightarrow b' : k) : t'$
- the modifier corresponding to

  $$\text{TSubstitution}(mr, r'/r, A(r') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r') = e : z) : t'$$

  assigns, to r at t, the rule $r' : (d_1 : s_1, \ldots, d_o : s_o \Rightarrow e : z) : t'$
- the expression Derogate$(mr, r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_m : z_m), \mathscr{F}(b' : k) : t'$ corresponds to applying two distinct modifiers. The first modifier assigns, to r at t, the rule $r' : (A(r) \cup \{q_1 : z_1, \ldots, q_m : z_m\} \Rightarrow b' : k) : t'$ The second assigns, to r at t, the rule $r'' : (A(r) \cup \{q_1 : z_1, \ldots, q_m : z_m\} \rightsquigarrow \sim b : j) : t'$.

Modifications	**Conditions**
Annulment$(mr, r) : t'$ PSubstitution$(mr', r, \mathscr{A}_r(a'_l : x_l/a_l : y_l, \ldots, a'_{l+m} : x_{l+m}/a_{l+m} : y_{l+m})) : t''$	$t' = t''$
Annulment$(mr, r) : t'$ PSubstitution$(mr', r, \mathscr{C}_r(b' : k/b : j)) : t''$	$t' = t''$
Annulment$(mr, r) : t'$ TSubstitution$(mr', r'/r, A(r') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r') = e : z) : t''$	$t' = t''$
Annulment$(mr, r') : t'$ TSubstitution$(mr', r'/r, A(r') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r') = e : z) : t''$	$t' = t''$
Annulment$(mr, r) : t'$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_m : z_m), \mathscr{F}(b' : k)) : t''$	$t' = t''$
Annulment$(mr, r') : t'$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_m : z_m), \mathscr{F}(b' : k)) : t''$	$t' = t''$
Annulment$(mr, r'') : t'$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_m : z_m), \mathscr{F}(b' : k)) : t''$	$t' = t''$
PSubstitution$(mr, r', \mathscr{A}_{r'}(a'_l : x_l/a_l : y_l, \ldots, a'_{l+m} : x_{l+m}/a_{l+m} : y_{l+m})) : t'$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_s : z_s), \mathscr{F}(b' : k)) : t''$	$t' = t''$ and $\exists u, v, l \leq u \leq l+m$ and $1 \leq v \leq s$ such that $a_u = q_v$
PSubstitution$(mr, r'', \mathscr{A}_{r'}(a'_l : x_l/a_l : y_l, \ldots, a'_{l+m} : x_{l+m}/a_{l+m} : y_{l+m})) : t'$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_s : z_s), \mathscr{F}(b' : k)) : t''$	$t' = t''$ and $\exists u, v, l \leq u \leq l+m$ and $1 \leq v \leq s$ such that $a_u = q_v$
TSubstitution$(mr, r'''/r, A(r''') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r''') = e : z) : t''$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_s : z_s), \mathscr{F}(b' : k)) : t''$	$t' = t''$
TSubstitution$(mr, r'''/r', A(r''') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r''') = e : z) : t''$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_s : z_s), \mathscr{F}(b' : k)) : t''$	$t' = t''$
TSubstitution$(mr, r'''/r'', A(r''') = \{d_1 : s_1, \ldots, d_o : s_o\}, C(r''') = e : z) : t''$ Derogate$(mr', r, r', r'', \mathscr{E}(q_1 : z_1, \ldots, q_s : z_s), \mathscr{F}(b' : k)) : t''$	$t' = t''$

Table 1. Conflicts.

3. Conflicts between Norm-modifications

Table 1 summarises all cases of basic conflicts between different norm modifi cations. Notice that in all cases a conflict obtains only if the conflicting modifi cations apply to the same time instant. Annulment of r is obviously incompatible with any partial substitution in r (fi rst and second rows from the top). The same applies to a total substitution which replaces r with r' when we also have that r or r' is annulled (third and fourth row). A similar intuition holds for the three subsequent rows: it is impossible to derogate to r if this last rule, or r' or r'', are dropped from the system. Exactly for the same reasons, derogation to r is incompatible with total substitution of r, or of r' or r'' (the fi rst three rows from the bottom). Finally, the cases in the fourth and fi fth rows from the bottom state that a partial substitution in the antecedent of a rule is incompatible with a derogation if at least one literal used in r' or r'' to derogate to r is replaced in r' or r''.

4. The Inference Machinery

A defeasible theory D is a structure $D = (\mathscr{T}, F, R^c, R^O, \prec)$ where $\mathscr{T}$ is discrete totally ordered set of instants of time, F is a fi nite set of temporalised literals, R^c is a fi nite set of temporal counts-as rules with viewpoint, R^O is a fi nite set of temporalised deontic rules with viewpoint, $\prec$, the superiority relation, is defi ned as $(R^c \times R^c \times \mathscr{T}) \cup (R^O \times R^O \times \mathscr{T})$. A conclusion in Defeasible Logic can have one of the following four forms:

$+\Delta@t\ q : t'$ meaning that q is defi nitely provable, at time t' with viewpoint t, in D (i.e., using only facts and strict rules).

[2]Again, to simplify the discussion, we will assume that r is defeasible and that r' will have the same strength of r. Of course, that r be defeasible is a necessary requirement, otherwise r'' could not block r.

$-\Delta@t\ q:t'$ meaning that we have proved that q is not definitely provable, at time t' with viewpoint t, in D.

$+\partial@t\ q:t'$ meaning that q is defeasibly provable, at time t' with viewpoint t, in D

$-\partial@t\ q:t'$ meaning that we have proved that q is not defeasibly provable, at time t' with viewpoint t, in D.

For example, $+\partial_O@t_1\ q:t_0$ means that we have a defeasible proof for $\boldsymbol{Obl}q$ at t_0, or, in other words, that $\boldsymbol{Obl}q$ holds at time t_0 when we use the rules in force in the normative system at time t_1. To specify whether a conclusion $q:t$ has been obtained via transient rules or via persistent rules we will introduce auxiliary proof tags indicating persistency or transiency. The proof tags are labelled with the mode used to derive the rule, according to their appropriate proof conditions. It is not possible to give the complete set of proof condition in this paper. Here we concentrate only on the proof conditions to derive defeasible persistence of both rules, and literals with both counts-as and obligation mode. The proof conditions given here are extensions of those given in [1]; the omitted proof conditions can be analogously obtained.

Provability is based on the concept of a *derivation* (or proof) in D. A derivation is a finite sequence $P=(P(1),\dots,P(n))$ of tagged literals satisfying the proof conditions (which correspond to inference rules for each of the kinds of conclusion). $P(1..n)$ denotes the initial part of the sequence P of length n. A strict derivation (i.e., a conclusion tagged with Δ) is just a monotonic derivation using forward chaining of rules, that is, modus ponens. In Defeasible Logic a defeasible derivation, on the other hand, has three phases. In the first phase we propose an argument in favour of the conclusion we want to prove. In the simplest case this consists of an applicable rule for the conclusion (a rule is applicable if the antecedent of it has already been proved). Then in the second phase we examine all possible counter-arguments (rules for the opposite conclusion). Finally we have to rebut the count-arguments. Thus we have to provide evidence against the counter-argument. Accordingly we can demonstrate that the argument is not as such (i.e., some of its premises are not provable), or we can show that the counter-argument is weaker than an argument for the conclusion. For persistent conclusions we have another method. We can use a derivation of the conclusion at a previous time provided that no terminating event occurred in between.

In [1] the rules are given, but the formalism we have introduced in the previous sections allows us to have rules in the head of counts-as rules, thus we have to admit the possibility that rules are not only given but can be derived. Thus in the proof conditions we have to cater for this option. Accordingly we have to give conditions that allow us to derive rules instead of literals. For the sake of simplicity we will assume that all rules in R can be overruled/modified. Then we have to extend the notation $R[x:t]$ to the case where x is a rule label (and norm-modifications). Given a set of (counts-as) rules R and a set of rule modifiers $M=\{m_1,\dots,m_n\}$, then

$$R[r:t_r]=\{s\in R: C(s)=m_i(v:t_v) \text{ and } m_i(v:t_v)=r:t_r\}$$

$R[r:t_r]$ gives the set of nested rules whose head results in the rule $r:t_r$ after the application of the rule modifier; and

$$R[\sim r:t_r]=\{s\in R: C(s)=m_i(r:t_r) \text{ and } m_i(r:t_r) \text{ is in conflict with } r:t_r\}$$

The set $R[\sim r : t_r]$ gives the set of rules that modify $r : t_r$ and the modification is in conflict with the $r : t_r$, see Table 1 for such conflicts. We can now give the proof condition for $+\partial^p$ to derive a rule.

If $P(n+1) = +\partial^p @t\ r : t_r$ then
1a) $r : t_r @t \in R^c$ or
1b) $\exists s : t_s \in R^c[r : t_r]$ such that $+\partial @t\ s : t_s \in P(1..n)$ and
  $\forall a : t' \in A(s), +\partial @t\ a : t' \in P(1..n)$; and
2) $\forall v : t_v \in R^c[\sim r : t_r]$ if $+\partial @t\ v : t_v \in P(1..n)$, then either
  2.1) $\exists b : t'' \in A(v)$ such that $-\partial @t\ b : t'' \in P(1..n)$ or
  2.2a) $v : t_v \prec_t r : t_r$ if 1a obtain or
  2.2b) $v : t_v \prec_t s : t_s$ if 1b obtain; or
3) $+\partial^p @t'\ r : t_r \in P(1..n)$, $t' < t$ and
  3.1) $\forall t'', t' \le t'' < t, \forall s : t_s \in R[\sim r : t_r]$ if $+\partial @t''\ s : t_s \in P(1..n)$, then either
    3.1.1) $\exists a : t_a \in A(s), -\partial @t''\ a : t_a \in P(1..n)$ or $t_s < t_r$; and
4) $+\partial^p @t\ r : t'_r \in P(1..n)$, $t'_r < t_r$ and
  4.1) $\forall t', t'_r \le t'' < t_r, \forall s : t_s \in R[\sim r : t_r]$ if $+\partial @t'\ s : t_s \in P(1..n)$, then either
    4.1.1) $\exists a : t_a \in A(s), -\partial @t'\ a : t_a \in P(1..n)$ or $t_s < t'_r$.

Let us briefly examine the above proof conditions. To prove a rule at time t, the rule must be in force at time t, i.e., the rule must be one of the given rules (condition 1a). There is a second possibility the rule is derived from another rule. The second rule must be provable and applicable at t (condition 1b). However, this is not enough since there could have been modifications to the rule effective at t. Thus we have to show that either all eventual modifications are not applicable (2.1) or the modifications are not successful since they are defeated (2.2a and 2.2b). Finally the rule could be provable because it was persistent, i.e., it was persistently in force before (3), and no modification occurred in between. The possible modifications in force after the rule was in force are not applicable to the rule. Or (4) the rule was persistently effective before, and its effectiveness was not revoked. The inference condition for positive persistent defeasible proofs is as follows.

If $P(n+1) = +\partial^p @t\ q : t'$ then
1) $+\Delta^p @t\ q : t' \in P(1..n)$, or
2) $-\Delta @t \sim q : t' \in P(1..n)$, and
  2.1) $\exists r : t_r \in R^p_{sd}[q : t']$: $+\partial @tr : t_r \in P(1..n), \forall a : t_a \in A(r : t_r), +\partial @t\ a : t_a \in P(1..n)$ and
  2.2) $\forall s : t_s \in R[\sim q : t]$: if $+\partial @t\ s : t_s$, then either $\exists a : t_a \in A(s : t_s), -\partial @t\ a : t_a \in P(1..n)$; or
    2.2.1) $\exists w : t_w \in R[q : t]$: $+\partial @t\ w : t_w \in P(1,,n)$ and
      $\forall a \in A(w : t_w), +\partial @t\ a : t_w \in P(1..n)$ and $w \succ s$; or
3) $\exists t'' \in \mathscr{T} : t'' < t$ and $+\partial^p @t''\ q : t' \in P(1..m)$ and
  3.1) $\forall t'''\ t'' < t''' \le t\ \forall s : t_s \in R[\sim q : t']$: if $+\partial @t'''\ s : t_s \in P(1..n)$, then either
    3.1.1) $\exists a : t_a \in A(s : t_s), -\partial @t'''\ a : t_a \in P(1..n)$ or
    3.1.2) $\exists v : t_v \in R[q : t'], +\partial @t'''\ v : t_v \in P(1..n)$ and
      $\forall b : t_b \in A(v : t_v) + \partial @t'''\ b : t_b \in P(1..n)$ and $s : t_s \prec_{t'''} v : t_v$; or
4) $\exists t'' \in \mathscr{T} : t'' < t'$ and $+\partial^p @t\ q : t'' \in P(1..m)$ and
  4.1) $\forall t'''\ t'' < t''' \le t'\ \forall s : t_s \in R[\sim q : t''']$: if $+\partial^p @t\ s : t_s \in P(1..n)$, then either
    4.1.1) $\exists a : t_a \in A(s : t_s), -\partial @t\ a : t_a \in P(1..n)$ or
    4.1.2) $\exists v : t_v \in R[q : t'''] + \partial @t\ v : t_v \in P(1..n)$ and
      $\forall b : t_b \in A(v : t_v) + \partial @t\ b : t_b \in P(1..n)$ and $s : t_s \prec_{t'''} v : t_v$.

The rationale of above proof conditions is the same of those in [1]. The main difference is that here every time we use a rule we have to verify that the rule is provable in the system. In addition we have to cater for both the persistence of the effectiveness of the rule (4) and that the rule was previously persistently in force (3).

5. Time-forking and Norm-modification

Retroactivity may be applied to basic rules as well as to norm-modifi cations. Cases of retroactivity of basic rules occur when we have rules such as $r : (a : x \Rightarrow^p_O p : y) : z$. r is a permanent (defeasible) deontic rule in force since z, persisting through subsequent instants $z_* \geq z$, stating that if a occurs at x then p is obligatory since time instant y. Suppose we have that $y = x+1$ and $z \geq 5$, and in particular that $z = 5$. If the theory contains $a : 2$, this allows for the derivation of $+\partial^p_O@5\ p : 3$, namely $\boldsymbol{Obl}p$ at time 3 from the viewpoint of 5, which is the time-instantiation of r being in force. At 5, in fact, r is in force; on the other hand, there is no time-constraint over x and y, and so a occurring at 2 permits to derive retroactively (namely, with respect to 5) p at 3. Suppose that the theory contains also

$$s : (b : j \Rightarrow^p_O \neg p : j) : k$$

such that $k = 2$. If $b : 3$ we would derive $+\partial^p_O@k\ \neg p : 3$. Since $k = 2$, then $+\partial^p_O@5\ \neg p : 3$ thus getting a conflict. If we want to have r prevailing over s the theory should say that $r \succ_t s$, $t = 5$. Notice, however, that we will still derive $+\partial^p_O@k_*\ \neg p : 3$ where $2 \leq k_* \leq 4$. This means that, if we have another rule

$$s' : (\boldsymbol{Obl}\neg p : i \Rightarrow^p_O q : i) : m$$

such that $m = 2$, then we will derive $+\partial^p_O@m_*\ \neg q : i$ where $2 \leq m_* \leq 4$ but $-\partial^p_O@m_*\ \neg q : i$ where $m_* > 4$. Let us see now cases of retroactivity of norm-modifi cations. Suppose r is defi ned as at the beginning of this section. Again, if we have $a : 2$, then $+\partial^p_O@5\ p : 3$. Imagine the system contains the following meta-rule, stating the annulment of r:

$$mr : (c : x' \Rightarrow^p \text{Annulment}(mr, r) : y') : z'$$

such that $z' = 4$ while $z = 3$ (3 is the time-instantiation of the being-in-force of r). This meta-rule assumes for its application that r is in force, which is true from time instant 3 on (rules persist over time unless cancelled or modifi ed). Let c be a future but uncertain event under which the norm-modifi cation will apply. This means that $x' > z$. However, we state that mr works retroactively, e.g., that $y' = 3$. Suppose we have $c : 7$. Accordingly, we will derive $+\partial^p@z'_*\ \text{Annulment}(mr, r) : 3$, where, e.g., $z'_* = 5$. As we have stated, a modifi cation is nothing but a function which assigns, to the modifi ed rule, another rule which holds at the time of the modifi cation and so at the time-instantiation of the modifi cation and at every subsequent time instant. In the case of annulment, the value of the the function is the empty rule, since the rule associated to the name r is dropped from the system. Thus, after the application of mr, we will also get $+\partial^p@z'_*\ (r : \perp) : y'$. Thus, from the viewpoint of $z'_* = 5$, if $a : 2$, this will no longer permit to derive the modalised consequent of r, i.e., $\boldsymbol{Obl}p : 3$ because, at that perspective, r does not exist anymore: $-\partial^p_O@z'_*\ p : 3$. Finally, notice that if mr were a transient rule, we would be able, under appropriate conditions, to model the notion of temporary annulment: in fact, in transient rules the effects are co-occurrent with the conditions. Logically, this is just a variation of the intuition just discussed. Let us see two cases, one of retroactive partial substitution and one retroactive derogation. Even here, dealing with these cases is a matter of variation of the previous logic intuitions. So we will only provide a brief discussion. Given r

as above, suppose we have the following meta-rule, stating the substitution, in the body of r, of $a : x$ with $d : x$:

$$mr' : (c : x' \Rightarrow^p \text{PSubstitution}(mr', r, \mathscr{A}_r(d : x/a : x)) : y') : z'$$

such that $z' = 4$ while $z = 3$. If we assume the same conditions of the case of the annulment just discussed, the fact of having $a : 2$ permits to derive $\boldsymbol{Obl}p : 3$ only if the viewpoint is such that $z_* < 4$. When mr' comes to be in force, the conditions for obtaining $\boldsymbol{Obl}p$ from r will be no longer a but d. Let us fi nally focus on the notion of retroactive derogation. Given r as above, suppose we have the following meta-rule, stating the derogation of r when the additional condition q obtains and such that the different effect p' should follow under this additional condition:

$$mr'' : (c : x' \Rightarrow^p_c \text{Derogate}(mr'', r, r', r'', \mathscr{D}(q : x), \mathscr{E}(p' : y) : y') : z'$$

Again, we assume the same conditions of the case of the annulment just discussed. Under those conditions we can reiterate the same argument and so we can derive $+\partial^p_O@z_*\ p : 3$ given $a : 2$. However, if mr'' is made applicable, we will derive $+\partial^p@z'_*$ Derogate$(mr, r, r', r'', \mathscr{E}(q : x), \mathscr{F}(p' : y)) : y'$, namely that this holds from the viewpoint of z'_* where $z'_* \geq 4$. According to the defi nition of Derogate, this conclusion is associated to adding in the theory the following two rules:

$$r' : (a : x, q : x \Rightarrow^p_O p' : y) : y' \qquad r'' : (a : x, q : x \leadsto^p_O \neg p : y) : y'$$

So, after the application of mr'', we will have r' and r'' in force since time instant 3. Thus, if we have $a : 2$ but also $q : 2$, then we will no longer derive $\boldsymbol{Obl}p$. In particular, at time 3, this conclusion will be blocked and we will get $+\partial^p_O@3\ p'$.

6. Summary

We extended the defeasible logic presented in [1] by allowing nested rules in the head of counts-as rules. This extension increases the expressive power of the logic and it allows us to represent meta-norms describing norm-modifi cations. We outlined the inferential mechanism needed to deal with the derivation of rules. Then we described some issues related to norm versioning and we illustrated the techniques with some relevant norm-modifi cations such as annulment, partial and total substitution and derogation. We showed that the formalism introduced here is able to deal with complex scenarios such as retroactivity and time-forking.

References

[1] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law*, pages 25-34, New York, ACM, 2005.

[2] J. Martinek and J. Cybulka. Dynamics of legal provisions and its representation. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law*, pages 20-24, New York, ACM, 2005.

[3] M. Palmirani. Time model in normative information systems. In *Post-proceedings of the ICAIL Workshop on the Role of Legal Knowledge in e-Governement*, Wolf, Tilburg, 2005.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Invited Address:

Making Sense of Coherence
The Level Theory of Coherence

Luc J. Wintgens
K.U. Brussels, Belgium

1. The Problem Framed

The use of logic in legal system has seduced a number of lawyers and logicians. Legal systems are thought to be expressible in logical language. The presupposition of this is that law is fit for logical treatment. Roughly two groups can be distinguished, the formalists and the non-formalists. The former hold that law is fit for logical or formal treatment, while the latter deny that.

The formalists are right in that at least some parts of the legal system can be expressed in formal language. The problem raised by their approach consists of "sinning by exaggeration". The belief in the possibility of some formalisation raises the hope that (1) the whole of a legal system can be formalised and, in a more abstract and stronger version that (2) any legal system can.

The non-formalists hold that formalisation is possible for parts of a legal system, or for some section of its operationalisation, though (1) not for the system as a whole and *a fortiori* (2) not for any legal system.

The formalist has a point in that the final result of the application of the rules of the system to a *casus* must, for reasons of justice, be similar in similar cases. If it were not, the legal system as a set of rules would not make sense. The rule character of law expresses a concern for equality as a core element of justice. That is why a legal system consists of general rules, the application of which does not depend on the officials dealing with them. The non-formalist approach for its part has a point in holding that the "application" of rules in chess or logic is different from "following rules" in normative action. The formalist and the non-formalist approach are opposed on at least this point. The former claim that consistency is a necessary condition for coherence while the latter hold that it is not.This paper argues for the non-formalist approach. According to a current view, the consistency of a legal system is a necessary condition for its coherence. A system then is coherent when it is completely consistent. Consistency and coherence are substitutable terms. It follows that a legal system containing one contradiction is not consistent, hence not coherent. This is an implausible conclusion, which is connected to the formalistic approach.

Therefore, I argue that the relation between consistency on the one hand and coherence on the other is in need of reconsideration. Consistency is not a necessary condition for coherence. Coherence is a form of "making sense". Consistency on its turn is a specific form of making sense. The absence of contradictions makes a set of propositions consistent. It does not follow from that that this set of propositions "makes sense" in some ordinary understanding of it. It may be coherent in a *specific* sense, that is, *free from contradictions*.

The foregoing points to a re-calibration of the relation between coherence and consistency in what I propose to call the “level theory of coherence”. This theory sets out the framework of a decision theory for not formalised sets of propositions, typically a legal system.

2. The Level Theory of Coherence Articulated

The level theory of coherence consists of four levels. The four levels are : the level of internal coherence; the level of consecutive coherence; the level of system coherence; and the level of environment coherence. Any form of rational discourse has to correspond to a level of coherence below which it makes no sense at all. This is the first level of coherence, the level of *internal* coherence, or coherence$_0$. The level of coherence0 is formally universal, in that it makes part of any form of rational discourse. It is though not materially universal in that it is not identical for every type of rational discourse.

The operationalisation of the existing set of propositions (that is, e.g., a legislator creates a new rule; an executive promulgates an executive order; a judge decides a case by giving his sentence) shows coherence in a different light. Judges are to decide cases by applying a rule or a precedent, which is required by formal justice. The same holds for the members of the executive or their administration. Yet, legislators cannot decide A and shortly after decide non-A. This requirement of stability in legislative ruling has a counterpart in judicial ruling where it is labelled “formal justice”. The common platform is a form of coherence called *consecutive* coherence, coherence over time, or the first level of coherence (coherence1).

System coherence or the level of coherence$_2$ comes into play when judges interpret rules systematically. This comes to read or interpret a rule (or principle) in the light or with the help of another rule (or principle). Legislators argue on the level of coherence$_2$ or system coherence when they take other rules of the legal system into account in order, e.g., not to annul the effects of existing rules without explicitly revoking them (typically: the unemployment trap). The level of coherence$_3$ or *environment coherence* comes into play when a legal system’s environment is taken into consideration. Judges and legislators typically use arguments like “values found in society”. This suggests that they lean over the edges of the legal system. In doing so, they import cognitive or normative contents into the legal system. In order for the legal system “to make sense as a whole”, that is, to be coherent, the rationality of this import operation is to be showed. The relation between a legal system and its “environment” has several and different aspects: the environment of a legal system is theoretical (or can only be known as a theory; this makes that legal interpretation (e.g. on the level of coherence2) is not (1) independent of or external to nor (2) purely internal to the legal system. It depends on a theory of language that makes the legal system make sense. The more general claim that is argued for is that the legal system is itself a theory that is dependent on other theories. This variant of the Gödel theorem in number theory is helpful in showing (1) the connection between different levels of coherence and (2) the possibility and the necessity of working with a level of coherence located “outside” the legal system. With the help of the level of coherence3, some problems of rule making and rule interpretation can be articulated and eventually be solved.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Mixing Legal and Non-legal Norms

Alexander Boer [1], Tom van Engers and Radboud Winkels
Leibniz Center for Law, University of Amsterdam, The Netherlands

Abstract. In Boer et al. (viz. [9]) we argued that evaluation of draft legislation, change from an old to a new regime, harmonization of legislation from multiple jurisdictions, and the decision to move a good, person, or service over the borders of a jurisdiction, involves a process of integration and comparison of preference structures. This paper argues that legal norms are in many contexts best understood as expressions of a ceteris paribus preference, and that this viewpoint adequately accounts for normative conflict and contrary-to-duty norms.

Keywords. Preference, Norms, Semantic Web, Decision Support Systems

1. Introduction

The interest in Legal Knowledge Representation used to be confined to small companies building legal expert systems. Customers of these companies were primarily interested in the expert system, while the accompanying knowledge representation was only used for validation purposes, if it was used at all. In recent years, however, we have seen increasing interest in the legal knowledge representation itself from large administrative organizations, and they want to use it for other purposes than building expert systems.

Knowledge representation is increasingly considered an enabling technology for complex administrative change processes driven by new legislation and policy guidelines. Our work for the Dutch Tax and Customs Administration (DTCA; cf. [9]) was clearly related to the huge change process triggered by the complete overhaul of the Dutch income tax law. DTCA knowledge representations were also expected to specify – and justify – how business processes were grounded in legislation, and provided input for the drafting process through formal verification and simulation of draft legislation. The *Juridisch Loket* (cf. [34]) project on pro bono legal assistance and the *DURP* project on spatial planning (cf. [8]) are also driven by an overhaul of legislation, and we have received inquiries from other administrative organizations involved in similar change processes.

The E-POWER project with the DTCA also involved exploratory research into using knowledge representations for comparing and harmonizing legislation (cf. [9]). This addresses a need that arises as EU agreements increasingly require public administrations to facilitate the movement of people, goods, and services over national borders. As these movements increase in frequency and importance, public administrations are increasingly confronted with the need to know about differences between legal systems in Europe. The EU INSPIRE initiative will create similar pressures in spatial planning, as

[1] Correspondence to: aboer@uva.nl

geospatial data about spatial legal abstractions will be made available throughout Europe but the meaning of these abstractions will remain as opaque as it was before.

In [9] we argued that evaluation of draft legislation, change from an old to a new regime, harmonization of legislation from multiple jurisdictions, and the decision to move a good, person, or service over the borders of a jurisdiction, involves the same underlying process of comparison of alternative norm systems.

Evaluation of legislation involves comparing the real world effects of a preference structure imposed by legislation against an external preference structure, imposed by so-called norms of analysis (viz. [21]). This is what allows choice between alternative codifications of political goals. We also noted that – judged against the norms of analysis – two very different legislative solutions, for instance taxing and prohibiting, can achieve the same policy goals because of complex interactions between legal norms and presumed non-legal preferences of the addressee of legislation. In both cases, the solution to this problem requires using legal and non-legal preference structures in a single decision problem.

As a side-effect, this view also accounts for rational planning and design subject to legal norms. In for instance [3], the role of legal norms in planning is restricted to pruning certain illegal plans from the alternatives being assessed. Legal constraints and constraints from other sources are applied in two separate stages, and usually in a different way, in assessment of alternatives. This approach unnecessarily complicates the process: by performing two separate assessments, a legal and a non-legal one, it does not for instance allow for weighing the consequences of violating a legal norm against the benefits of complying with other non-legal preferences. Rational violation of the law is not generally perceived as a valuable functionality by public administration, but this perceived gap between decision theory and legal theory disqualifies legal knowledge representation as exotic and hard to understand.

This begs the question as to whether it is possible to account for the ontological connection between legal norms and non-legal norms. [6] Considers the ramifications of doing so in detail. This paper is a summary of part of that work. This paper argues that legal norms are in many contexts better understood as expressions of a ceteris paribus preference, and that this viewpoint adequately accounts for normative conflict and contrary-to-duty norms.

1.1. A Workbench for Legal Knowledge Engineering

Deontic concepts are not easily combined with industry standards like UML and OWL. We experienced this when in the DTCA a UML CASE tool was used, even though this created major problems for representing normative knowledge (cf. [7]). Large organizations are however not easily convinced to abandon "commercial standards" or widely used CASE tools, if only because of a, generally justifiable, perception of lack of continuity and quality of technical support of more experimental systems.

Legal knowledge representation also proves hard to sell to knowledge engineers with a more general IT background. In reality deontic representation is probably not necessary for most legal expert systems in institutional settings, as we also noted in [10]. We recently reviewed a knowledge representation of a part of social security law made by a company which was building a system for the organization that issues permits for dismissing personnel. Outcome of the review was that there were no normative conflicts in

the set, and the distinction between permissions and obligations of the organization in the relevant social security laws was immaterial because all individual discretion of civil servants was taken away by the internal guidelines of the organization.

We did establish that the knowledge representation was too specific to be reusable in another institutional setting where the *same* legislation is used from another point of view[1].

The Leibniz Center for Law is starting a new EU-funded project, ESTRELLA, to define a standard for exchange of legal knowledge in XML. In addition, we are planning to build a new knowledge engineer's workbench, informally called Newton for now, in association with a number of industrial and governmental partners. The added value of the new standard is largely determined by the potential for reuse of knowledge in different settings. Reuse is a necessary condition for business models based on content syndication: if a knowledge representation of some legal source is reusable in different institutional contexts, then legal knowledge bases can be produced and maintained by third parties and bought off the shelf. We have therefore been working on concepts of tools for automated ontology integration, integration of normative systems, and support for the comparison of normative systems (viz. [10,7,9]).

We realized that designing business processes and comparing legislation both involved the problem of mixing legal and non-legal norms in one decision problem. The law is not the only constraint on a suitable business process design. Norms however take different shapes in different domains. How they interrelate is best understood by understanding all types of norms as preferences between classes of alternatives in a decision problem.

2. The Ontological Status of Norms

From research in the past (viz. [14,11,13]), it has become clear that knowledge representations are not generally automatically reusable outside the specific context for which the knowledge representation was originally developed. Part of this problem has been attacked with ontologies. Ontologies are however only reusable if they were explicitly designed to be reusable, which is generally not the case: Breuker and Hoekstra argued in [12] that Legal Knowledge Engineering suffers from epistemological promiscuity, or indiscriminately mixing epistemological knowledge and domain knowledge in ontologies.

Knowledge about reasoning – epistemology – and knowledge about the problem domain – domain ontology – are to be separated if the knowledge representation is to be reusable. Mixing the two categories of knowledge leads to confusion in reuse (cf. [14]) because both types of knowledge are to be reused in different contexts: reasoning methods are specific for types of tasks (planning, design, diagnosis, assessment, etc.), but independent of the type of domain (cf. [11,13]), while domain knowledge is used – in different ways – in different types of tasks.

The *norm* is an epistemological concept identified by its *role* in a type of reasoning and not something that exclusively *belongs* to the vocabulary of the legal domain. A norm is a standard of performance, a measurement scale that defines what is normal or normative. It is used whenever we are assessing something, regardless of domain.

[1] The expert system described in [34] for a legal assistance agency that i.a. assists people in appealing against an unfair dismissal.

The norm is conceptualized differently in different contexts: When assessing readings from a broken instrument – for instance some circuit board – a norm is for instance the distribution of scores obtained from a correctly functioning norm group, or some margin around the specified ideal functional mapping from input to output. In sociology, a *social norm* is a pattern of behavior expected of an agent within a particular society in a given situation. Only within this context – the behavior of a supposedly rational agent is being assessed – the norm is interpreted as reflecting some preference of the involved agent; given that an agent is aware of a number of behavioral alternatives and made a choice between them, we can infer that the agent revealed a *preference* since choice and preference are interdefinable. If we observe a pattern of agents revealing the same preference, we may infer that having the preference is *normal*.

The *legal norm* is an extension of the social norm in a specific kind of social system. Directing others to adopt preferences is the core business of legislators. Since legal norms are usually communicated by prose, the preferences that the legislator communicates to us are understood as a system of declarations and directives (obligations, permissions, and prohibitions). This is by no means the only way to direct others: teachers usually grade the work of students for instance, which is a declarative act, and in this act implicitly direct others. Note that a declaration differs considerably from an assertion in the illocutionary sense (cf. [30]). The latter expresses a belief, and is constative, while the former brings something about, i.e. is performative.

The distinction between declarative and directive acts is very similar to the distinction between constitutional and regulative norms made by Boella and Van der Torre in [5]. They observe that constitutional norms can be *likened* to beliefs, and regulative norms to goals. Epistemologically this is defensible: regulative norms are to treated as one's goals, and constitutive norms are to be adopted as if they are beliefs when reasoning about how these regulative goals apply to one's life.

We prefer to think of the directive act as an attempt to make the reader commit to a preference. The term 'goal' suggests that the preference is absolute, and not *ceteris paribus* (cf. [16]) as is normally the case with adopted social norms. Goals are better understood as the minimized set of propositions to be achieved in a certain situation according to a preference structure, which may contain conflicting preferences. Constitutive norms are supposed to *overrule* one's own beliefs, and regulative norms to overrule one's own preferences. In [10] we sketched a legislative feedback loop: the citizen evades negative consequences of legislation by refining his preferences, or by violating norms, and the legislator bases legislation on historical knowledge of, or assumptions about, what the citizen will do.

In this paper we will attempt to ground norms from the legal field in a more general epistemological concept from other fields of research that deal with assessment and planning. Representing norms as preferences has clear pragmatic benefits. Firstly, preference and utility are widely understood by knowledge engineers and computer scientists outside our field, while deontic operators are not. Secondly, grounding deontic operators in preference makes it possible to compose complex preference structures relating legal and non-legal preferences, which is a necessity in planning, design, and Comparative Law.

3. Norms as Preference Statements

An agent complies with a norm by knowing of it, adopting it, and revealing it in his choices. The agent violates a norm by making choices in contradiction with the norm. A norm is a ceteris paribus preference (cf. [16]) for some type of alternative over another type of alternative, given that one is otherwise indifferent between those alternatives.

Let β be a situation, and α an alternative in a menu[2]. The deontic operators are reduced to preference statements as follows:

$$
\begin{aligned}
O(\alpha|\beta) &= \boxed{\beta,\alpha} \prec \boxed{\beta,\neg\alpha} \\
F(\alpha|\beta) &= \boxed{\beta,\neg\alpha} \prec \boxed{\beta,\alpha} \\
P(\alpha|\beta) &= \boxed{\beta,\alpha} \preceq \boxed{\beta,\neg\alpha}
\end{aligned}
$$

The reduction of obligations – and prohibitions – follows the reduction applied by for instance [4]; This view is implicit in existing preference-based solutions to the contrary-to-duty problem discussed in section 3.1. The straightforward reduction of permissions is new, as far as we know.

The permission is treated as a statement of qualified indifference ($\alpha \preceq \neg\alpha$). Note that some broad permissions can be interpreted as symmetrical ($\alpha \sim \neg\alpha$), and there is evidence, i.a. from psychology (viz. [23]), that the common sense interpretation of permission is indeed symmetric. This idea is the notion of *strong indifference* of i.a. Opalek and Wolenski in [27]. In law, it is for instance generally accepted that freedom of expression includes the freedom to keep your opinion to yourself.

This view has been discarded as wrong by i.a. Alchourron and Bulygin in [1], and compatibility with existing theories of deontic reasoning demands the asymmetry. The permission only becomes meaningful if it interferes with one or more obligations or prohibitions, and is used in a reasoning system that chooses between incompatible deontic statements and for some reason prefers the permission. The asymmetric statement ($\alpha \preceq \neg\alpha$) leaves room for a prohibition or obligation ($\neg\alpha \prec \alpha$) without creating a conflict, and retains the information that the permission was about α and not $\neg\alpha$. The reduction respects $O(\alpha|\beta) \equiv F(\neg\alpha|\beta) \equiv \neg P(\alpha|\beta)$ and $P(\alpha|\beta) \equiv \neg O(\neg\alpha|\beta)$.

In the next subsections we review the consequences of the treatment of legal norms as preferences for the representation of contrary-to-duty sets of norms, and for detection of normative conflicts. Because subsumption relations play a key role in this discussion, we will introduce a graphical convention for displaying an entailment relation $\mathsf{KB}, \alpha' \vdash_s \alpha$ (e.g. a bicycle α' is a vehicle α) in deductive system S using knowledge base KB:

$$
\begin{array}{c}
\boxed{\alpha} \\
\uparrow \\
\boxed{\alpha'}
\end{array}
$$

We have to introduce another notational convention to make clear that a statement $(\beta,\alpha) \prec (\beta,\neg\alpha)$ is an ordering on α and $\neg\alpha$ within the context of β (or $\beta \Rightarrow (\alpha \prec \neg\alpha)$ following [15]):

[2]Beware of interpreting α as an action: the alternatives may concern having a bilge pump or a bilge ejector for disposing of bilge on an existing ship. Key is that the situation can be remedied by an agent.

$$\beta : \boxed{\alpha} \prec \boxed{\neg\alpha}$$

3.1. Contrary-to-Duty Obligations

The contrary-to-duty (CTD) obligation arises in a sub-ideal situation, brought about by the violation of a primary obligation. The relation between the primary obligation and CTD-obligation can be represented according to the following schema:

$$\begin{tikzcd} \boxed{\alpha} & \prec & \boxed{\neg\alpha} & \\ & \boxed{\neg\alpha, \beta} \arrow[ur] & \prec & \boxed{\neg\alpha, \neg\beta} \arrow[ul] \end{tikzcd}$$

The CTD-obligation in the bottom row only distinguishes cases in which $\neg\alpha$ is already the case, in both proscribed and prohibited situations. The CTD-obligation should not suggest a permission to enter into the sub-ideal situation, although there are real world cases where they are explained that way.

This type of situation should not be considered merely theoretical. All norms regulating punishment by the legal system, contractual remedial and reparational obligations, and liabilities arising from contract violations follow the CTD-obligation pattern.

The Chisholm paradox is an instructive example of how to analyze complex sets of logical relations between primary obligations and CTD-obligations. Since we are not building a logic we are obviously not dealing with a paradox, but merely representing the situation that gives rise to it. The Chisholm set consists of the following norms:

1. $O(\alpha|\top)$
2. $O(\beta|\alpha)$
3. $O(\neg\beta|\neg\alpha)$

The paradox arises in standard deontic logic (SDL) when α is the case. α reads "a man goes to the assistance of his neighbours", and β reads "the man tells his neighbours that he will come".

Since the first preference expresses a preference for both (β, α) and $(\neg\beta, \alpha)$, the imposed ordering – from most to least ideal – is the desired one (cf. [33]):

$$\boxed{\alpha, \beta} \prec \boxed{\alpha, \neg\beta} \prec \boxed{\neg\alpha, \neg\beta} \prec \boxed{\neg\alpha, \beta}$$

The Reykjavic and Forrester sets are other interesting variations that have been analyzed this way. Time adds an interesting twist to the Chisholm situation; The choice for β can only be realized before the choice for α. A more straightforward and intuitive temporal example, by Prakken and Sergot in [28], admonishes one to keep one's promises, and to apologize if one doesn't. None of these present us with any problems.

Instead of viewing a CTD-obligation as an obligation arising from the sub-ideal situation in which one norm has already been violated, one could see the CTD-obligation as arising from the violation itself in some cases, as Governatori et al. (cf. [19]) seem to do. Arguably this is for instance the case in the following set:

1. One ought not to break a promise.
2. One may break a promise if one pays a 100$ fee to the government for a permit to break a promise.

3. If item 1 is violated, then one ought to apologize.

The question is whether item 1 is still violated if one pays the fee. Does one still need to apologize? Governatori et al. (cf. [19]) solve this by the introduction of an operator $O(\alpha) \otimes O(\beta)$, which is read as "$O(\beta)$ is the reparation of the violation of $O(\alpha)$". For us this is an open problem.

3.2. Conflicts between Norms

The legislator does not usually include pairs of directly opposite norms like $O(\alpha)$ and $P(\alpha)$, but pairs $O(\alpha)$ and $P(\alpha')$ where α' is subsumed by α, or $\alpha' \vdash \alpha$, do occur quite regularly.

An instrumental concept in explaining conflicts between norms is that of *realizability*; A norm is *realized* if the state of affairs it permits or mandates is the case. There is a conflict between a pair of norms if they are not *jointly realizable*. In terms of preferences, conflicts surface as circularities. An important point is that conflicts or circularities between norms do not depend on any case at hand (e.g. cf. [35,4]).

A distinction is usually made between so-called conflicts of disaffirmation and compliance conflicts. Lindahl (cf. [24]) defines disaffirmation as follows: "a relation between two norms of different deontic mode, one being permissive and the other mandatory".

A disaffirmation conflict is in our context a circularity between a permission and either an obligation or prohibition. One such situation – a conflict of disaffirmation between $O(\alpha|\beta)$ and $P(\neg\alpha'|\beta')$ where $\alpha' \vdash \alpha$ and $\beta' \vdash \beta$ – can be displayed as follows:

$$\begin{array}{ccccc} \beta : & \boxed{\quad \alpha \quad} & \prec & \boxed{\quad \neg\alpha \quad} \\ \uparrow & \uparrow & & \downarrow \\ \beta' : & \boxed{\quad \alpha' \quad} & \succeq & \boxed{\quad \neg\alpha' \quad} \end{array}$$

We will call this a *disaffirmation of an imperative*. Intuitively it is meant as an exception and takes precedence to the primary obligation, although this is not necessarily the case. In analogy, we have a *disaffirmation of a permission*. Disaffirmation allows the legislator to create exceptions to provisions easily.

There are two other disaffirmation conflicts that have no intuitive solution, or clear purpose. The following set of simple rules exemplifies this type of case:

1. Using the network facilities in the classrooms is prohibited.
2. Using WiFi in the university building is permitted.

It is not clear which of the two rules is an exception to the other one, granted that we share the belief that using WiFi is subsumed by using network facilities, and being in the classrooms is subsumed by being in the university building. Intuitively, the most specific description of the alternatives, in this case using WiFi, takes precedence. Arguably, these sets belong to Hill's "intersection" conflicts (cf. [17,22]).

The compliance conflict is more complicated and the length of this paper does not allow for good explanation, but the representation in terms of preference structures does distinguish the relevant types in a satisfactory way. This type of set – a *conflict of compliance* between $O(\alpha)$ and $O(\neg\alpha')$ where $\alpha' \vdash \alpha$ – can be displayed as follows:

$$\begin{tikzcd} \boxed{\alpha} \arrow[r, phantom, "\prec"] & \boxed{\neg\alpha} \arrow[d] \\ \boxed{\alpha'} \arrow[u] \arrow[r, phantom, "\succ"] & \boxed{\neg\alpha'} \end{tikzcd}$$

An interesting real-world example is when the Amsterdam police ordered nightclub owners to lock the emergency exits to keep drugs out, while the fire department ordered the same nightclub owners to unlock them to allow for escape in case of disaster.

Other classes of perceived conflict concern situations where a combination of norms defeats the underlying norms of analysis, as is the case in a situation involving two permissions in Elhag et al. ([17]), and situations where the realization of a norm is impossible per se or does not depend on any agent choice, as is the case when the legislator for instance prohibits fatal accidents.

4. Composition of Ceteris Paribus Preferences

Contrary to normal preference structures that are revealed by choices, ceteris paribus preference structures may turn out to contradict eachother (viz. [16]). If there is a cicularity in the ceteris paribus preference structure, then the condition that we are otherwise indifferent between the alternatives has been violated, and we compose a new norm.

Hansson ([20]) distinguishes between preferences among incompatible alternatives, which he refers to as *exclusionary* preferences, and preferences among compatible alternatives, which he refers to as *combinative* preferences. When one chooses a red car over a blue car, one is exhibiting an exclusionary preference, because the alternatives are incompatible. A preference for having a cheap car rather than a red car, by contrast, is combinative, because the alternatives are compatible.

Legal norms are all examples of a particular type of statements of exclusionary preference. The norm does not even explicitly specify the alternative: it is the logical complement of that which ought to be done, or left undone. A conflict between two norms is always solved by choosing between the offending preferences, instead of weighing them.

A non-legal preference may take all kinds of forms: it may impose an ordering with more than two places, and it may be combinative. A underspecified norm of analysis $\alpha \prec \beta$ (economic growth is better than wasting taxpayer's money, for instance) may for instance end up being realized in the form $\alpha \prec \neg\alpha$ and $\neg\beta \prec \beta$.

Conflicts between non-legal preferences may be resolved with all kinds of weighing procedures (e.g. cf. [15]). Choice is a rare solution: the most common ones are additive value function and the multiplicative utility function, even when the required preference independence between the involved preferences is doubtful, as is for instance the case in contrary-to-duty situations. Choice can be considered a weighing procedure where the value of alternative behaviors is determined solely by the most preferred preference among the conflicting ones. Because composition functions for non-legal preferences, and pairs of legal and non-legal preferences, can take any shape, it is not possible to design a single choice rule for conflict resolution that is always valid.

Choice rules for legal reasoning are possible, because there are well-established principles for choosing between legal norms. Such rules were explicitly specified in [32,35], and preference-based or defeasible deontic logics obviously also contain a choice rule even if it is not explicitly separated from the logic itself.

4.1. Composition of Legal Norms

Composition of legal norms is fairly straightforward, because the legal system uses a composition procedure that is very intuitive. It can be understood as an analogy of belief revision.

We all know that beliefs can sometimes be wrong, so intelligent beings need to be able to revise beliefs when they acquire new information that contradicts their old beliefs. Reasoning systems modeling this phenomenon are called *belief revision* systems. One common way of determining *which* beliefs should be surrendered is to use a so-called *epistemic entrenchment* ordering (cf. [25,18]). This ordering expresses the idea that some of our beliefs are more fundamental than others; It is a preference ordering on beliefs.

Makinson and Gardenförs ([18]) showed that there is a tight connection between belief revision and defeasible reasoning: belief revision models temporal nonmonotonicity, while default reasoning models logical nonmonotonicity.

Norms define an analogous preference ordering on norms, that helps us assess things. Over time we refine our norm systems. In legal theory we find the principles of Lex Posterior, the newer norm is preferred over the older one, and Lex Specialis, the more specific norm is preferred over the newer one. Any provision obviously has a history and propositional content. A Lex Specialis ordering between two provisions can be discovered by comparing the logical content of the provisions. A Lex Posterior ordering between two provisions can be discovered by comparing the history of the provisions.

The Lex Specialis and Lex Posterior principles *describe* certain phenomena of reasoning in general. They do not sanction the preference of the newer over the older, or the specific over the general, but merely observe that it *is* so. These principles work in any legal system, regardless of whether they are codified. Codification does make it harder to overrule them in lower legislation, though.

The third wellknown principle, Lex Superior, is a fundamentally different type of principle. The Lex Superior principle is also ‘intuitive’, but it is a design principle for complex, hierarchical legal systems. This hierarchy is not something which is discovered by application of the principle to a pair of provisions: it has been designed into the legal system. Provisions overruling the normal activity of Lex Specialis and Lex Posterior usually define some hierarchical preference ordering on legal sources that can be used for choosing between norms if they form a cyclical preference.

Principles like Lex Posterior are sometimes cancelled or amended by legislation. This is fairly rare, but Suber ([31]) collected a valuable collection of real world examples. A cancellation example of Lex Posterior is for instance the so-called ‘entrenched’ rule that explicitly prohibits its own amendment. The legislator has to resort to a specific prohibition because it is not possible to explicitly ‘overrule’ an unstated principle. These examples can be easily represented as orderings dictated by Lex Superior: the entrenched rule is in a Lex Superior category declared by itself.

The logic of the legal system dictates that Lex Superior must normally take precedence over Lex Specialis and Lex Posterior. The relative priority of the Lex Posterior and Lex Specialis principles is however in principle unsettled. The reason for this is that they are implicitly assumed to reinforce eachother: assuming that the legislator refines his expressed preferences over time, is aware of his own acts in the past, and intends it’s new provisions to be compatible with the existing corpus, it is only reasonable to expect that new preference statements refine the existing system.

A special case is where the Lex Posterior ordering exists, but a Lex Specialis ordering does not. This is the case in a situation of symmetric disaffirmation. To this situation there is no satisfactory prima facie solution, because the legislator is able to directly repeal the offending older provision and yet failed to do so. This suggests that the legislator did not explicitly intend the ordering that is imposed by Lex Posterior. Still one usually favors the ordering imposed by Lex Posterior if there are no other clues, because the choice between norms must be made[3].

5. Discussion

It is surprising that the similarity between the norms in assessment and the preferences of decision theory is rarely exploited in Computer Science & Law. Computer Science & Law tends to stay close to the surface structure of legislation, and somehow wants to capture the *speech act*, instead of the expressed preference. In most applications of norms in Computer Science & Law the epistemological mechanism by which norms are adopted is however completely irrelevant, and it is usually not accounted for (but see for instance [29] for a notable exception).

It is interesting to note that this view on norms is discarded when dealing with case law. In the context of case-based reasoning it is acknowledged that many deontic norms that distinguish between only two valuations (legal–illegal) can be considered indicative of the existence of a more fundamental preference scale underlying it. In court it is often possible to argue successfully that this underlying scale represents the actual intent of the legislator and can be taken into account (e.g. cf. [2]). If the legislator for instance orders civil servants to always buy the cheapest of some good, that statement is indicative of a more general preference for cheaper goods over more expensive goods if one is otherwise indifferent between the alternatives.

The citizen and judge are supposed to solve conflicts using the rules in section 4.1. Legislation is designed so that these rules work. The legislator, on the other hand, can use more refined strategies for combining preferences during the design of legislation. If this is the case, one wonders why the legislator is not more direct in the expression of his preferences. The best explanation is that continuous scales become too complex to handle in case of a conflict between several competing values, although we have noted that this arguably happens in court rooms anyway. Little empirical research has been invested in the question which way of describing the desirable behaviour actually works best (cf. [26]). The purpose of traditions, guidelines, manuals, and computer programs is to enforce a consistent style of drafting, and consistency in style is obviously one possible way to increase interpretive clarity.

As noted in section 4.1 there are situations that *feel* like a normative conflict, but can only be explained as a conflict by involving underlying preferences. The scenario mentioned by Elhag et al. ([17]) involves two permissions:

> There seem to be other types of conflict as that between the permission for A to live in a certain house and a permission for B to destroy that same house. These conflicts need our attention and have to be embodied in a theory on normative conflicts.

[3] Using whatever criterium. For example, one U.S. court held that when two amendments adopted at the same time are irreconcilable, the one receiving the greater number of votes in the Senate takes priority over the other (cf. [31]). This exotic principle has not been accepted in jurisprudence anywhere, as far as we know.

With our analysis on evaluation of legislation (viz. [9]) in mind, we hope to build a knowledge representation language in ESTRELLA in which this conflict can finally be expressed.

References

[1] C. Alchourron and E. Bulygin. The expressive conception of norms. In E. Hilpinen, editor, *New studies in Deontic Logic*, pages 95–124. D. Reidel, Dordrecht, 1981.

[2] T. Bench-Capon and G. Sartor. Using values and theories to resolve disagreement in law. In J. Breuker, R. Leenes, and R. Winkels, editors, *Legal Knowledge and Information Systems (JURIX-2000)*, Amsterdam, 2000. IOS Press. ISBN 1.58603.144.9.

[3] G. Boella, L. Lesmo, and R. Damiano. On the ontological status of norms. In *Law and the Semantic Web*, volume 3369 of *Lecture Notes in Computer Science*. Springer Verlag, 2005.

[4] G. Boella and L. W. N. van der Torre. Permissions and obligations in hierarchical normative systems. In *ICAIL*, pages 109–118, 2003.

[5] G. Boella and L. W. N. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, Whistler (CA), 2004.

[6] A. Boer. *Representation and Evaluation of Legislation*. PhD thesis, University of Amsterdam, Forthcoming, Amsterdam, 2006.

[7] A. Boer and T. van Engers. A Knowledge Engineering Approach to Comparing Legislation. In *Proceedings of the Conference on Knowledge Management in Government*, Rhodes, Greece, 2003. Springer Verlag.

[8] A. Boer and T. van Engers. Standards for spatial regulations. In *Proceedings of the Workshop on the Role of legal knowledge in e-Government, held in conjunction with the Tenth International Conference on Artifi cial Intelligence and Law*, Bologna, Italy, 2005.

[9] A. Boer, T. van Engers, and R. Winkels. Using Ontologies for Comparing and Harmonizing Legislation. In *Proceedings of the International Conference on Artifi cial Intelligence and Law (ICAIL)*, Edinburgh (UK), 2003. ACM Press.

[10] A. Boer, T. van Engers, and R. Winkels. Normative statements on the semantic web. In *Proceedings of the Workshop on Legal Ontologies and Artifi cial Intelligence Techniques, held in conjunction with the Tenth International Conference on Artifi cial Intelligence and Law*, Bologna, Italy, 2005.

[11] J. Breuker. A suite of problem types. In J. Breuker and W. V. de Velde, editors, *CommonKADS Library for Expertise Modelling*, pages 57 – 88. IOS-Press/Ohmsha, Amsterdam/Tokyo, 1994.

[12] J. Breuker and R. Hoekstra. Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law. In *Proceedings of EKAW Workshop on Core ontologies*. CEUR, 2004.

[13] B. Chandrasekaran. Generic tasks in knowledge-based reasoning: Characterizing and designing expert systems at the 'right' level of abstraction. In *Proceedings of the IEEE Computer Society Artifi cial Intelligence Applications Conference*. IEEE Computer Society, December 1985.

[14] W. J. Clancey. The epistemology of a rule-based expert system: A framework for explanation. *Artifi cial Intelligence*, 20:215–251, 1983.

[15] J. Doyle and R. H. Thomason, editors. *Working papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*. American Association for Artifi cial Intelligence, Menlo Park, California, 1997.

[16] J. Doyle and M. P. Wellman. Representing preferences as ceteris paribus comparatives. In *Decision-Theoretic Planning: Papers from the 1994 Spring AAAI Symposium*, pages 69–75. AAAI Press, Menlo Park, California, 1994.

[17] A. A. Elhag, J. A. Breuker, and B. W. Brouwer. On the formal analysis of normative conflicts. In H. van den Herik et al., editor, *JURIX 1999, The Twelfth Conference*, Legal Knowledge Based Systems, pages 35–46, Nijmegen, 1999. GNI.

[18] P. Gärdenfors and D. Makinson. Revisions of knowledge systems using epistemic entrenchment. In M. Y. Vardi, editor, *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the Second Conference*, pages 83–96. Morgan Kaufmann, Los Altos, CA, 1988.

[19] G. Governatori and A. Rotolo. Modelling contract using ruleml. In T. Gordon, editor, *Legal Knowledge and Information Systems (Jurix 2004)*, pages 141–150, Amsterdam, 2004. IOS Press.

[20] S. O. Hansson. *The Structure of Values and Norms*. Cambridge University Press, UK, 2001.

[21] W. Hettich and S. L. Winer. Rules, Politics, and the Normative Analysis of Taxation. Carleton Economic Papers 00-12, Carleton University, 2000.

[22] H. Hill. A functional taxonomy of normative conflict. *Law and Philosophy*, 6:227–247, 1987.

[23] M. Keller, M. Gummeran, X. Wang, and S. Lindsey. Understanding perspectives and emotions in contract violation: Development of deontic and moral reasoning. *Child Development*, 75:614–635, 2004.

[24] L. Lindahl. Conflicts in systems of legal norms: A logical point of view. In P. Brouwer, B. Hol, A. Soeteman, W. van der Velde, and A. de Wild, editors, *Coherence and Conflict in Law*, pages 39–64, Deventer, 1992. Kluwer.

[25] B. Nebel. Syntax-based approaches to belief revision. In P. Gärdenfors, editor, *Belief Revision*, volume 29, pages 52–88. Cambridge University Press, Cambridge, UK, 1992.

[26] V. Nourse and J. Schacter. The Politics of Legislative Drafting: A Congressional Case Study. *New York University Law Review*, 77:3:575–624, 2003.

[27] K. Opalek and J. Wolenski. Normative systems, permission and deontic logic. *Ratio Juris*, 4:334–348, 1991.

[28] H. Prakken and M. Sergot. Dyadic deontic logic and contrary-to-duty obligations. In *Defeasible Deontic logic: Essays in Nonmonotonic Normative Reasoning*, volume 263 of *Synthese library*. Kluwer Academic Publisher, 1997.

[29] G. Sartor. Why agents comply with norms, and why they should. In R. Conte and C. Dellarocas, editors, *Social Order in MAS*, Dordrecht, 2002. Kluwer.

[30] J. Searle and D. Vanderveken. *Foundations of illocutionary logic*. Cambridge University Press, Cambridge, England, 1985.

[31] P. Suber. *The Paradox of Self-Amendment: A Study of Law, Logic, Omnipotence, and Change*. Peter lang Publishing, New York, 1990.

[32] A. Valente and J. Breuker. ON-LINE: An architecture for modelling legal information. In *International Conference on Artifi cial Intelligence and Law (ICAIL-1995)*, pages 307–315, 1995.

[33] L. W. N. van der Torre and Y. Tan. The temporal analysis of Chisholm's paradox. In T. Senator and B. Buchanan, editors, *Proceedings of the Fourteenth National Conference on Artifi cial Intelligence and the Ninth Innovative Applications of Artifi cial Intelligence Conference*, pages 650–655, Menlo Park, California, 1998. AAAI Press.

[34] T. M. van Engers, R. Winkels, A. Boer, and E. de Maat. Internet, portal to justice? In T. Gordon, editor, *Legal Knowledge and Information Systems. Jurix 2004: The Seventeenth Annual Conference*, pages 131–140, Amsterdam, 2004. IOS Press.

[35] R. Winkels, D. Bosscher, A. Boer, and J. Breuker. Generating Exception Structures for Legal Information Serving. In T. Gordon, editor, *Proceedings of the Seventh International Conference on Artifi cial Intelligence and Law (ICAIL-99)*, pages 182–195, New York (NY), 1999. ACM.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Regulations Expressed As Logical Models (REALM)

Christopher Giblin [a], Alice Y. Liu [b], Samuel Müller [a,1], Birgit Pfitzmann [a] and Xin Zhou [b]

[a] *IBM Zurich Research Lab, Switzerland*
[b] *IBM China Research Lab, China*

Abstract. Recent years have seen a number of high-profile incidents of corporate accounting fraud, security violations, terrorist acts, and disruptions of major financial markets. This has led to a proliferation of new regulations that directly impact businesses. As a result, businesses, in particular publicly traded companies, face the daunting task of complying with an increasing number of intricate and constantly evolving regulations. Together with the growing complexity of today's enterprises this requires a holistic compliance management approach with the goal of continually increasing automation.

We introduce REALM (Regulations Expressed as Logical Models), a metamodel and method for modeling regulations and managing them in a systematic lifecycle in an enterprise. We formalize regulatory requirements as sets of compliance rules in a novel real-time temporal object logic over concept models in UML, together with metadata for traceability. REALM provides the basis for subsequent model transformations, deployment, and continuous monitoring and enforcement of compliance in real business processes and IT systems.

Keywords. Compliance Management, Regulatory Requirements, Logical Formalization, Temporal Logics, MDA

1. Introduction

Recent years have witnessed a growing amount of regulatory requirements directed towards businesses, particularly publicly traded companies. Prominent examples of such regulations include the Sarbanes-Oxley Act, the U.S. Patriot Act, Basel II, anti-money laundering regulations and various de facto standards such as the International Financial Reporting Standards (IFRS). Not only the sheer number of relevant laws and standards but also the complexity of individual regulations is drastically increasing. Consequently, affected businesses are confronted with the task of adapting to new and evolving regulatory requirements. While this process is initially driven by regulators, companies increasingly recognize this challenge as an opportunity to improve operational transparency, traceability and reporting.

We propose a systematic compliance management approach for addressing the increasing breadth and complexity of regulatory requirements on businesses. Our approach

[1]Correspondence to: Samuel Müller, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland. Tel.: +41 44 724 8275; Fax: +41 44 724 8953; E-mail: sml@zurich.ibm.com.

is based on a lifecycle view of compliance management, which begins with the formalization of regulations by means of metamodel for regulatory requirements that we call REALM. Figure 1 provides a visual representation of the metamodeling approach as compared with addressing each regulation as a single case. With the latter method, a worst case $M \times N$ mappings are required to implement M regulatory requirements over a set of N target systems. Systems include business processes, IT applications and systems, and people issues such as education material. In contrast, the metamodeling approach theoretically involves only $M + N$ transformations.

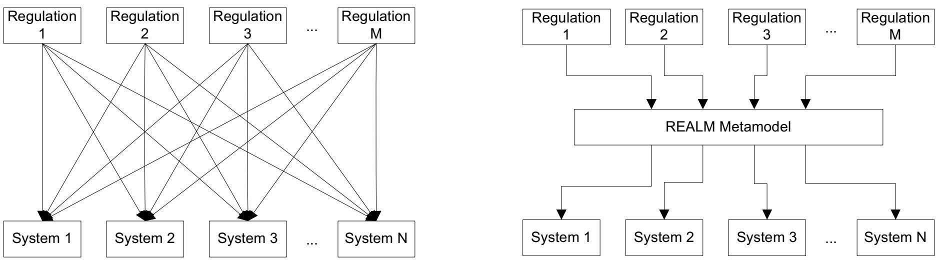

Figure 1. Advantage of using the REALM metamodel for business compliance.

The benefit of the REALM metamodel is that different regulations, to a large extent, can be captured and formalized using a shared language and semantics. The reuse of model elements and tools further eases the understanding and enforcement of regulatory requirements. An additional advantage of systematic compliance management is that it offers traceability from the regulations to the systems; this enables businesses to demonstrate to auditors or regulators how they achieve compliance, which is increasingly required by new regulations.

A REALM model consists of the following parts:

- *Concept model.* We represent the domain of discourse of a regulation by a concept model expressed in a UML profile developed for REALM.
- *Compliance rule set.* Regulatory requirements are represented as a set of formulae in a novel real-time temporal object logic over the elements from the concept model. Thus, e.g., sequencing and time constraints on actions can be expressed.
- *Metadata.* We capture meta-information about the structure of the legal source as well as lifecycle data such as the enactment date.

The main novelty of our work is our use of a temporal object logic for regulation modeling. The emphasis on temporal aspects stems from our focus on proactively enforcing regulations at the level of business processes. In contrast, prior work focusses mainly on comparing laws, applying them to finished cases, and on legal discourse, so that temporal aspects are of lesser specific importance. We need a real-time logic, i.e., a logic that captures exact times and intervals and not only the order of events, since many regulations contain concrete time constraints, such as data retention duration, the obligation to respond to requests within a certain time, or the requirement to perform certain actions contingent on other actions occurring or not within a specific past period. We are not aware of the prior use of a real-time temporal object logic in regulation modeling, and the logic itself is new in its combination of real-time temporal aspects with a modern object model.

REALM models of regulatory requirements are part of and support a compliance management lifecycle. Naturally, a regulation of interest must be initially stored, analyzed and understood. REALM models consisting of a concept model, a compliance rule set, and metadata can then be derived. These abstract models are ultimately transformed to implementation artifacts such as business process definitions, data retention policies, access control lists, or correlation rules, while preserving traceability to relevant passages in the respective regulation sources. The artifacts are deployed into the business and IT infrastructure of the enterprise. Compliance with the regulatory requirements can be monitored and enforced leveraging the capabilities of the involved target technologies and the common models shared by all stages of the compliance management lifecycle.

2. Related Work

Regulatory ontologies and logics have been commonly used in the development of expert and formal dialogue systems and more recently in compliance-assistance solutions [1, 2,3,4]. Exemplary models of concrete regulatory requirements include the formalization of the British Nationality Act, Dutch tax law and the Ontario Freedom of Information and Protection of Privacy Act, and the representation of environmental regulations [5, 6,7,8]. While we have found that regulations usually contain many timing requirements with references to real time, we are not aware of any efforts to formalize regulations concentrating on real-time temporal object logics. In contrast to other work addressing time in law (such as [9] or [10]), the real time temporal object logic of REALM integrates naturally with modern object-oriented modeling systems and is therefore more intuitive to use and understand in a business environment.

UML, the Unified Modeling Language, which we use for the concept models, is a widely used standard for modeling object-oriented systems and has also been used for ontology specification [11]. UML profiles are an extension mechanism for defining domain-specific modeling elements by specializing UML metaclasses.

Using the Object Constraint Language (OCL), UML models can be complemented by precise constraints. The official UML and OCL standards, however, provide no inherent support for expressing temporal predicates. Research into temporal extensions to the OCL includes [12,13,14]. However, these extensions seem not capable of specifying absolute real-time references as often needed in a legal context and may be impractical when global constraints over independent objects or over objects with non-intersecting lifetimes are needed. Furthermore, [14] does not support real time at all.

Propositional temporal logics (PTL) have mainly been proposed for the specification of dynamic systems such as programs [15]. For real-time aspects, we build upon the Timed Propositional Temporal Logic (TPTL), which employs a novel restricted use of temporal quantification, by introducing a so-called freeze quantifier that binds a variable to the time of a particular formula evaluation [16]. Approaches in logic to combine temporal and object aspects do not allow real-time features as we need them [17,18].

3. REALM-based Compliance Management Lifecycle

In order to enhance an enterprise's ability to adapt to new regulations and standards, we propose a compliance management lifecycle process based on REALM models. We

describe this lifecycle for one enterprise and one new regulation. The process can be abbreviated if another party, e.g., a standards body, has already formalized the regulation.

1. *Determination of scope.* First, the scope of the given regulation is determined and its relevance and potential impact on the enterprise are evaluated. This may restrict which parts of a given regulation must be modeled formally. It may also suggest which parts of the enterprise need to be analyzed in greater detail for compliance with this regulation.
2. *Formalization of regulatory requirements.* Next, those portions of the regulation in scope for the enterprise are formalized into a REALM model, i.e., a concept model, a compliance rule set, and metadata as described in the introduction. This process will typically produce two models:

   (a) *Immediate model.* First one creates a model that stays close to the terms and requirements of the regulation; we call this the immediate model.
   (b) *Refined model.* Many terms in a regulation are vague or coarse-textured, obliging the enterprise to choose an instantiation in line with current best practices and its own strategic goals, practices and compliance management objectives. We call the result of this phase, in which specific deployment values are selected, the refined model.

   Both immediate and refined models are REALM models, based on the same metamodel.
3. *As-is analysis of the enterprise.* In order to assess the impact of the selected regulatory requirements in detail, a thorough as-is analysis is needed. The results are models of the parts of the enterprise in scope of the regulation, e.g., business processes, applications, data models, and IT resources.
4. *Gap analysis.* Given rigorous and concise representations of the regulation and the potentially affected parts of the enterprise, the impact on the existing processes, data and resources is assessed by comparing the new REALM models with the as-is situation.
5. *Deployment.* Informed by the gap analysis the REALM models are deployed into the identified target systems. Even where no gaps currently exist, formal deployment may prevent new gaps from arising when processes or IT systems change in the enterprise. Depending on the type of requirements captured, target systems can be process definitions, access control lists, privacy policies, storage policies or correlation rules for event monitoring. The models are mapped onto these artifacts by means of model transformations. These diverse deployment procedures are represented in Figure 2. In terms of the Model Driven Architecture (MDA) [19], one can regard the REALM models as platform-independent models (PIM) and the target models as platform-specific models (PSM).
6. *Compliance monitoring and enforcement.* After a REALM model has been deployed into a target system, compliance is sometimes ensured automatically. For instance, a business process execution engine ensures that activities are executed in the order specified in the process definition; hence ordering constraints deployed by adapting that process definition are automatically enforced. However, due to the inherent possibility of human or system error, compliance must almost always also be monitored and enforced in real time. For instance, a business process definition may state that an activity only takes two days, but unexpected cir-

cumstances may hold it up. Hence a rule with a time constraint would also be deployed into a correlation engine for monitoring. Furthermore, if the correlation engine detects a timeout, its only means to resolve the issue may be to send an alert to the responsible compliance officer; in such cases compliance is finally enforced on a higher level and not always with fully predetermined procedures.

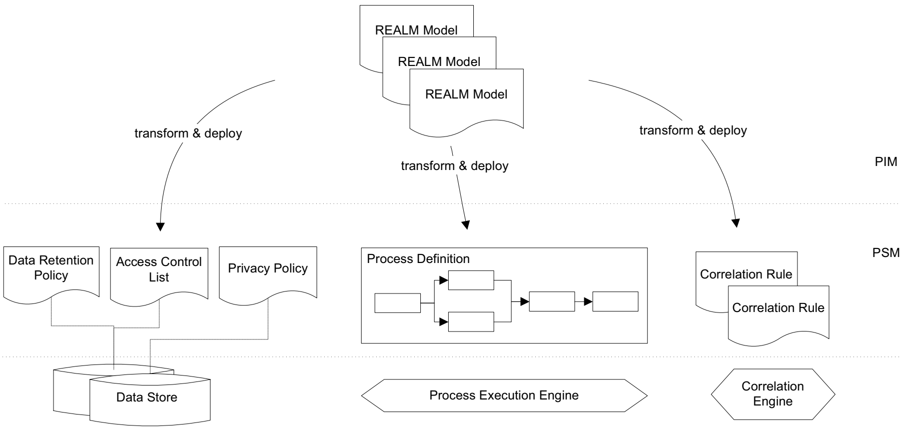

Figure 2. Exemplary model transformations for deployment.

The need for a refined model containing specific assignments from a set of possible instantiations of vague terms is a characteristic of modeling regulations for enterprise deployment. This contrasts to deciding compliance of finished cases, legal discourse, or law comparison, which can all be done with the immediate model alone. For instance, if a regulation states that a response to certain requests must be given within a "reasonable time", then an enterprise has to decide how fast it can provide such answers and how much earlier it wants to start an internal escalation process. Hence the immediate model contains the concept "reasonable time", while the refined model contains the concrete time assignment, e.g. "2 days". Retaining both models provides traceability in case the interpretations of or best practices for the vague term change, or the enterprise later reconsiders its initial decision, as in the case of our example of 2 days.

Whether the refinements are compliant is ultimately a legal interpretation for which an enterprise typically involves internal or external legal consultation. Capturing regulations formally does not obviate this need. However, formalizing a regulation as immediate and refined models facilitates identifying where such advice is needed and where decisions are applied. This enhances transparency by moving interpretation out of the hands of individual process owners or system administrators.

4. REALM Metamodel

In order to formalize regulations in the necessary detail, we need a language that is expressive enough to capture the variety of requirements occurring in actual regulations. We formalize the objects and relationships occurring in a regulation in a concept model (i.e., a domain ontology) and the constraints on these concepts by rules in a real-time

temporal object logic. The structural information of the actual regulations is captured with a metadata model.

4.1. Concept Model

The REALM concept model captures the concepts and relationships occurring in a regulatory domain. For example, a regulation requiring that *'Banks must verify the identity of each customer'* includes the concepts *bank*, *verify*, *identity*, and *customer* and implicit relations such as between *bank* and *verify* and between *verify* and *identity*.

REALM provides a set of predefined abstract types such as *Person* and *Organization*, *Process* and *Action*, *Artifact* and *Resource*, *Principle* and *Purpose*, as well as *Location* and *Cost*. We also include certain basic types such as String, Integer, and Boolean as in UML. Concrete concepts like *Customer* should, wherever possible, be subtypes of the predefined types, here *Person*. Individual instances like specific customers of a specific bank are instances of these types. Using the predefined abstract types, together with the following predefined relations, has two main benefits: easier and less error-prone construction of the following temporal formulae, and easier specification of automated model transformations as needed for deployment.

The concrete syntax of the predefined types and relations is given by a UML profile. In a regulation model based on this profile, a class stereotype denotes that a concept is of a certain predefined type and association stereotypes denote predefined relations. Our choice to make actions a separate predefined type in REALM, although object methods might seem more natural in the context of UML, was made with deployments in mind: Top-level deployments will often be to business process or workflow models, where actions are also separate.

Relations in regulations can be understood in the same way as in logic and UML. An equivalent term in logic is predicate; related terms in UML are associations and association classes. REALM provides several predefined relations over the predefined types. For example, *Do* is a predefined abstract relation between a person or organization and an action. We write it $Do(a,b)$, where a is an instance of a subtype of *Person* or *Organization* and b an instance of a subtype of *Action*. This predicate evaluates to true if a executes action b at the point in time where the predicate is evaluated. Other important examples of predefined relations are $On(a,b)$, $Input(a,b)$, and $Output(a,b)$ where a is an action and b an artifact; they have the natural meanings. Similar natural relations have been defined between most pairs of the predefined types.

Many time constraints in laws refer to the beginning or the end of an action. We use subscripts '$_S$' and '$_F$' for this. In the standard case of a relations with one action parameter, we attach these subscripts to the relations, e.g., $Do_S(a,b)$ and $Do_F(a,b)$ evaluate to true only at the point in time where the person or organization a starts executing action b or finishes it, respectively.

For readability of the later formulae, we define a syntactic scheme for composing predefined binary relations into n-ary ones: For relations $R_1,\ldots,R_n$, we define $R_1\cdots R_n(a_1,\ldots,a_{n+1})$ as $R_1(a_1,a_2)\wedge\ldots\wedge R_n(a_n,a_{n+1})$, where $R_1\cdots R_n$ is the string concatenation of the relation names. Clearly this is only defined if the second parameter of R_i has the same type as the first parameter of R_{i+1} for $i=1,\ldots,n-1$. Our naming scheme of predefined relations allows unique parsing of the concatenated names by the initial capitals. For instance, we can now write $DoOn(bank, open, account)$ for $Do(bank, open)\wedge On(open, account)$.

4.2. Compliance Rule Set

Compliance rules in REALM are expressed using a real-time temporal object logic. Recall that this logic as such is an important novel aspect of REALM. A REALM compliance rule set is based on a REALM concept model.

We employ the so-called definitional approach (cf. [20]) to the formalization of regulatory requirements because we focus mainly on the temporal aspects at this stage and do not intend to cover contrary-to-duty type obligations.

Structural requirements typically do not need a rule in the REALM compliance rule set because they can be expressed with existing features of UML within the concept model, in particular with multiplicities. For instance, one can thus model that a bank must have exactly one auditing committee, or that each account has a personal identification record attached to it.

We build upon the Timed Propositional Temporal Logic from [16]. The advantage of this treatment of real time is a good balance between expressiveness and complexity [21]. We add the object model instead of atomic propositions, the specialization to actions with lifetimes, and syntactic abbreviations such as the use of time with different units. The basis are timed state sequences. The following definition assumes a definition of a set States of states of our concept model, which we have only sketched above, and builds on a basic discrete time type Time (e.g., milliseconds).

Definition 4.1 (Timed State Sequence) *A timed state sequence* $\rho = (\sigma, \tau)$ *over a REALM concept model consists of a state sequence* $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ *and a time sequence* $\tau = \tau_0\ \tau_1\ \tau_2\ldots$ *with* $\sigma_i \in \mathsf{States}$ *and* $\tau_i \in \mathsf{Time}$ *for all* $i \geq 0$. *The time sequence must be monotonic, i.e.,* $\tau_i \leq \tau_{i+1}$ *for all* $i \geq 0$, *and provide progress, i.e., for all* $t \in \mathsf{Time}$ *there exists* $i \geq 0$ *such that* $\tau_i > t$. $\diamond$

Temporal formulae are based on state formulae, i.e., formulae evaluated on one state σ_i of a timed state sequence. State formulae are based on the relations from the concept model, with the usual logical connectives and with OCL-like navigation among the objects. For instance, a notation *a.customer* for an account instance *a* refers to the customer owning account *a* if there is an association from class *Accounts* to class *Customer* with multiplicity 1. We model instances as passivated after the end of their lifetime, but still available for navigation.

Temporal formulae are built with the usual (not real-time) temporal modalities such as $\Box$ (always), $\Diamond$ (eventually) and $\blacklozenge$ (sometimes in the past), as well as so-called freeze quantifiers and relations on times. A freeze quantifier corresponds to the introduction of a variable for a point in time. For instance, a formula part $\Diamond t.\phi$ means that eventually formula ϕ will hold, and we introduce the time variable t for this point in time. Time relations within ϕ can refer to t, e.g., to express that t is at most 2 days later than some other time t', introduced similarly with a freeze quantifier. Examples are given in Section 5.

As an example of how temporal modalities are defined, we present the definition of the always modality without freezing. The fact that formula ϕ is true for the timed state sequence ρ is denoted by $\rho \models \phi$. The state sequence ρ^i for $i \geq 0$ is defined by deleting the first i elements from ρ. Then $\rho \models \Box\phi \;:\Longleftrightarrow\; \forall i \geq 0 : \rho^i \models \phi$ for every formula ϕ. (Overall this is a definition by structural induction.)

Banks must implement procedures for verifying the identity of each customer; these procedures must ensure that the bank knows the true identity of each customer. At a minimum, the bank must obtain the following information prior to opening an account:
1. *Name;*
2. *date of birth;*
3. *residential address;*
4. *identification number.*

The bank must verify the identity of each customer, using the information obtained in accordance with the above requirements, within a reasonable time after the account is opened.
The bank must also implement procedures for responding to circumstances in which the bank cannot ensure that it knows the true identity of the customer. These procedures should describe when the bank should close an account, after attempts to verify the customer's identity have failed.
The bank must implement procedures for making and retaining a record of all information obtained according to the above requirements.
The bank must retain the recorded information for five years after the date the account is closed.

Figure 3. Fictitious regulation text inspired by 31 CFR paragraph 103.121 (Shortened for better readability).

4.3. Metadata

Besides concept models and compliance rule sets, REALM models may include metadata providing information about the modeled regulatory requirements. Two kinds of metadata are required:

- *Structural metadata* link the formalized regulatory requirements to their source. For instance, REALM model elements may be annotated with the name of the regulation, the modeled paragraph or section, a plain text description, and a hyperlink to an online source or interpretation.
- *Lifecycle constraints* include the enactment date of the modeled regulation, the validity duration, the expiry date, or validity constraints of individual compliance rules or model elements.

Relating REALM model elements to structural metadata is fundamental to the traceability and comprehensibility of formalized regulatory requirements.

REALM does not specify a single legislation metamodel but integrates with existing models such as MetaLex [22], PAPI [23] and EnAct [24]. MetaLex, e.g., provides lifecycle metadata for regulation parts through attributes such as *date-enacted*, *date-effective* and *date-repealed*.

5. Example

Figure 3 presents an example text that we capture using the REALM metamodel. It closely resembles the latest requirements made on financial institutions under the U.S. Patriot Act, Section 326, in particular the rule that implements it, 31 CFR paragraph 103.121.

We now formalize these regulatory requirements. The REALM concept model is shown in Figure 4. Recall that the concrete syntax is a UML class diagram using the REALM UML profile, with stereotypes denoting predefined types and relations. Let us explain a few non-obvious choices in this concept model: We model the retention of an account as an explicit action because it may involve active steps, such as ensuring

continued accessability, even if the software systems change in the five years after the account is closed. We model names etc. as attributes of both the customer and a record. The former denote the true attributes of this customer, the latter the information that the customer provides but that may still need verification. The attribute *successful* of the action type *VerifyIdentity* is by default initialized to false and becomes true if the correctness of the identity was established to the satisfaction of the bank.

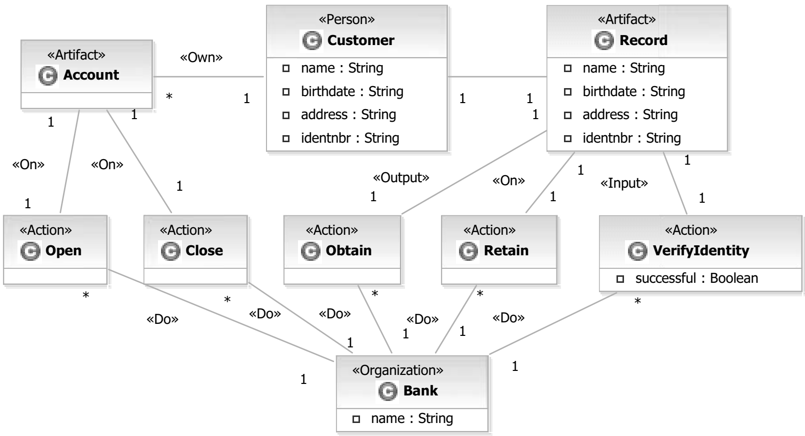

Figure 4. REALM concept model for the example regulation.

In the following, we present the REALM compliance rule set. Figure 5 illustrates the timing constraints on the actions for one account. Some details will become clear when the formulae are explained. Note that this is not an activity diagram; all the relations between actions which have no explicit constraints can be different than drawn here. One can in principle turn every set of constraints into an activity diagram with many alternatives, but as typical regulations consist of individual constraints, it is natural and useful for traceability to start by modeling the given constraints one by one as formulae. We plan to use time constraint diagrams as patterns for formula editing in the future. There is a certain similarity to the graphical interval logic of [25].

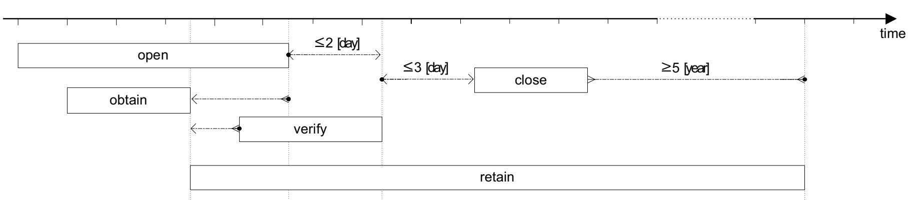

Figure 5. Time constraints on the involved actions for one account.

$$
\begin{aligned}
&\forall bank \in Bank,\ open \in Open, \exists obtain \in Obtain,\ r \in Record:\\
&\Box\ (Do_F(bank, open)\\
&\qquad \rightarrow \blacklozenge\ (DoOutput_F(bank, obtain, r) \qquad\qquad (1)\\
&\qquad\qquad \wedge\ r = open.account.customer.record \wedge\ r.name \neq null\\
&\qquad\qquad \wedge\ r.birthdate \neq null \wedge\ r.address \neq null\ \wedge\ r.identnbr \neq null))
\end{aligned}
$$

Formula (1) expresses the requirement that before a new account is opened, the bank must at least obtain the name, the date of birth, the address and an identification number of the customer. As to action durations, we required that the end of the action *obtain* is before the end of *open*. According to the concept model, the information is gathered in a record; we require that all four elements are non-null at the end of the action *obtain*. The fact that the record is about the customer opening the account is reflected by the equality about *r*; here we used OCL-like navigation. The temporal structure of the formula can be read as follows: "Whenever (□) the bank finishes opening an account, then sometimes in the past (⧫) it finished obtaining a record such that ..."

$$\begin{aligned} &\forall bank \in Bank,\ open \in Open,\ a \in Account\ \exists verify \in VerifyIdentity: \\ &\Box t_{open}.\ (DoOn_F(bank, open, a) \\ &\quad \rightarrow \Diamond t_{verify}.\ (DoInput_F(bank, verify, a.customer.record) \wedge\ t_{verify} - t_{open} \leq\ 2_{[day]})) \end{aligned} \tag{2}$$

Formula (2) states that whenever a bank has opened a new account eventually it has to finish the verification of the identity of the customer, based on the data collected in the process of opening the account. The formula further formalizes that the verification action needs to be completed within reasonable time, which has been interpreted to mean at most 2 days in the given example. Thus this formula belongs to the refined model in the sense of Section 3.[1]

$$\begin{aligned} &\forall bank \in Bank,\ a \in Account,\ verify \in VerifyIdentity: \\ &\Box\ t_{verified}\ ((DoInput_F(bank, verify, a.customer.record) \wedge\ verify.successful = false) \\ &\quad \rightarrow\ \Diamond\ t_{closed}.\ (DoOn_F(bank, close, a) \wedge\ t_{closed} - t_{verified} \leq\ 3_{[day]})) \end{aligned} \tag{3}$$

Formula (3) models that a bank must respond to circumstances where it cannot successfully verify a customer's identity and has to close a tentatively opened account. In our refined model we require that this happens within three days.

$$\begin{aligned} &\forall bank \in Bank,\ obtain \in Obtain\ \exists retain \in Retain: \\ &\Box\ (Do_F(bank, obtain)\ \rightarrow\ DoOn_S(bank, retain, obtain.record)) \end{aligned} \tag{4}$$

Formula (4) expresses that a bank has to retain the record of obtained customer information.

$$\begin{aligned} &\forall bank \in Bank,\ a \in Account,\ retain \in Retain\ \exists close \in Close: \\ &\Box\ t_{delete}.\ (DoOn_F(bank, retain, a.customer.record) \\ &\quad \rightarrow\ \blacklozenge\ t_{close}.\ (DoOn_F(bank, close, a) \wedge\ t_{delete} - t_{close} \geq\ 5_{[year]})) \end{aligned} \tag{5}$$

Formula (5) contains the constraint that a customer record may only be deleted if the account has been closed for at least 5 years.

$$\begin{aligned} &\forall bank \in Bank,\ c \in Customer,\ r \in Record,\ verify \in VerifyIdentity: \\ &\Box\ ((DoInput_F(bank, verify, r) \wedge\ verify.successful = true\ \wedge\ c = r.customer) \\ &\quad \rightarrow\ (r.name = c.name\ \wedge\ r.birthdate = c.birthdate \\ &\qquad \wedge\ r.address = c.address\ \wedge\ r.identnbr = c.identnbr)) \end{aligned} \tag{6}$$

Finally, formula (6) states that if the bank considers a verification successful then the record entries are the correct customer data. We also made this a temporal formula be-

[1]Recall that "." after a temporal operator and a variable is part of freeze quantification conserving the evaluation time. Otherwise it is OCL-like navigation.

cause the example regulation does not require that the bank continually watches whether the real customer data change. This requirement, in contrast to the previous ones, cannot be deployed in the strict sense, because the real customer data are out of the enterprise's scope of control. We formalize it nevertheless in the immediate model for traceability of the informal measures to be taken.

To conclude the picture, Figure 6 depicts an instance model of how a REALM compliance rule set can be annotated with metadata.

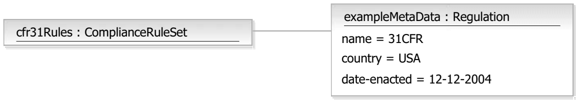

Figure 6. Annotating REALM models with metadata.

6. Conclusion

We have introduced REALM, a metamodel for formally expressing regulatory requirements with special emphasis on the use in proactive compliance management in enterprises. A REALM model of a regulation consists of three pillars: a concept model of the terms in the regulation, a compliance rule set in a novel real-time temporal object logic, and metadata designating the source regulation and validity dates. Using an example based upon the U.S. Patriot Act, we demonstrated that temporal aspects are important and that a real-time logic is needed. Temporal aspects are also the first aspects that need to be considered when adapting business processes. REALM is embedded into the larger context of an integrated compliance management process, which tracks regulatory requirements across their entire lifecycle, lends itself to model-driven transformation and deployment and allows for continuous compliance monitoring and enforcement.

Acknowledgments

We thank Günter Karjoth, Luke O'Connor, Matthias Schunter, Naishin Seki, Markus Stolze, Akihiko Tozawa, Michael Waidner, and the anonymous referees for their valuable contributions and comments, and June Y. Felix, Mark Greene, and Jürg von Känel for overall support.

References

[1] L. T. McCarty. A language for legal discourse – I. Basic Features. In *Proc. 2nd Int. Conf. on Artifi cial Intelligence and Law (ICAIL '89)*, 180–189. ACM Press, 1989.

[2] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *Proc. of JELIA'2000, The 7th Workshop on Logic for Artifi cial Intelligence*, Lecture Notes in AI, 224–238. Springer, 2000.

[3] H. Ryan, P. Spyns, P. De Leenheer, and R. Leary. Ontology-based platform for trusted regulatory compliance services. In *On the Move to Meaningful Internet Systems 2003: OTM Workshops 2003*, *Lecture Notes in Computer Science 2889*, 675–689. Springer, 2003.

[4] V. Kabilan, P. Johannesson, and D. M. Rugaimukamu. Business contract obligation monitoring through use of multi tier contract ontology. In *On the Move to Meaningful Internet Systems 2003: OTM Workshops 2003*, *Lecture Notes in Computer Science 2889*, 690–702. Springer, 2003.

[5] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–386,1986.

[6] T. M. van Engers, R. Gerrits, M. Boekenoogen, E. Glassée, and P. Kordelaar. Power: using UML/OCL for modeling legislation - an application report. In *Proc. 8th Int. Conf. on Artifi cial Intelligence and Law (ICAIL '01)*, 157–167. ACM Press, 2001.

[7] C. Powers, S. Adler, and B. Wishart. EPAL translation of the The Freedom of Information and Protection of Privacy Act, March 2004. IBM and Information and Privacy Commissioner/Ontario, `http://www.ipc.on.ca/docs/EPAL%20FI1.pdf`.

[8] S. Kerrigan and K. H. Law. Logic-Based Regulation Compliance-Assistance. In *Proc. 9th Int. Conf. on Artifi cial Intelligence and Law (ICAIL '03)*, 126–135. ACM Press, 2003.

[9] L. Vila and H. Yoshino. Time in automated legal reasoning. *Information and Communications Technology Law*,7(3):173–197, 1998.

[10] R. Hernandez Marin and G. Sartor. Time and norms: a formalisation in the event-calculus. In *Proc. 7th Int. Conf. on Artifi cial Intelligence and Law (ICAIL '99)*, 90–99, 1999.

[11] S. Cranefi eld and M. Purvis. UML as an ontology modelling language. In *Proc. of the Workshop on Intelligent Information Integration*, *16th Int. Joint Conf. on Artifi cial Intelligence ((IJCAI-99))*. CEUR Publications, 1999.

[12] S. Flake and W. Mueller. An OCL extension for real-time constraints. In *Object Modeling with the OCL*, Lecture Notes in Computer Science 2263, 2002. Springer.

[13] S. Flake and W. Mueller. Past- and future-oriented time-bounded temporal properties with OCL. In *Proc. of the 2nd Int. Conf. on Software Engineering and Formal Methods (SEFM 2004)*. IEEE Computer Society Press, 2004.

[14] P. Ziemann and M. Gogolla. An OCL extension for formulating temporal constraints. Technical Report 1/03, Universität Bremen, 2003.

[15] A. Pnueli. The temporal logic of programs. In *Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, 46–57. IEEE Computer Society Press, 1977.

[16] R. Alur and T. A. Henzinger. A really temporal logic. *Journal of the ACM*, 41/1:181–204, 1994.

[17] A. Sernadas, C. Sernadas, and J. F. Costa. Object specifi cation logic. *Journal of Logic and Computation*, 5(5):603–630, 1995.

[18] D. Distefano, J.-P. Katoen, and A. Rensink. On a temporal logic for object-based systems. In *4th Int. Conf. on Formal Methods for Open Object-Based Distributed Systems IV*, 305–325. Kluwer Academic Publishers, 2000.

[19] J. Miller, J. Mukerji (ed.). MDA guide version 1.0., May 2003. omg/2003-06-01, `http://www.omg.org/cgi-bin/doc?omg/03-06-01`.

[20] A. J. I. Jones and M. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artifi cial Intelligence and Law*,1(1):45–64, 1992.

[21] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Information and Computation*, 104/1:35–77, 1993.

[22] E-POWER Consortium. Metalex, version 1.02, June 2003. `http://www.metalex.org`.

[23] F. Zeni, S. McGrath, and C. Hatter. Pan african parliamentary interoperability (PAPI), Report and documentation, 2004. United Nations Department of Economic and Social Affairs, `http://www.parliaments.info/PAPI/`.

[24] T. Arnold-Moore and J. Clemes. Connected to the law: Tasmanian legislation using EnAct. *Journal of Information Law and Technology*, (1), 2000.

[25] L. K. Dillon, G. Kutty, L. E. Moser, P. M. Melliar-Smith, and Y. S. Ramakrishna. A Graphical Interval Logic for Specifying Concurrent Systems. *ACM Transactions on Software Engineering and Methodology*, 3(2):131–165, 1994.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Assumption Based Peg Unification for Crime Scenario Modelling [1]

Jeroen Keppens [a] and Burkhard Schafer [b]
[a] *King's College London, Dept. of Computer Science, Strand, London WC2R 2LS, UK*
[b] *The University of Edinburgh, School of Law, Edinburgh EH8 9YL, UK*

Abstract. An important cause of miscarriages of justice is the failure of crime investigators and lawyers to consider important plausible explanation for the available evidence. Recent research has explored the development of decision support systems that (i) assist human crime investigators by proposing plausible crime scenarios explaining given evidence, and (ii) provide the means to analyse such scenarios. While such approaches can generate useful explanations, they are inevitably restricted by the limitations of formal abductive inference mechanisms. Building on work presented previously at this venue, this paper characterises an important class of scenarios, containing "alternative suspects" or "hidden objects", which cannot be synthesised robustly using conventional abductive inference mechanisms. The work is then extended further by proposing a novel inference mechanism that enables the generation of such scenarios.

Keywords. Crime Investigation, Truth Maintenance, Abduction, Peg Unification

1. Introduction

In a previous Jurix paper [9], we have discussed largely informally the logic that governs hypothetical reasoning with evidence in a criminal law context. This paper builds on the results developed there, and addresses several points made by the then referees, in particular the desirability of a fully explicit formal account of that we had proposed. Hypothetical reasoning plays an important part in crime investigation and also in criminal trials. At the same time, difficulties with hypothetical reasoning strategies during crime investigations have been identified as a common source for miscarriages of justice [6].

While our focuses in the past has been primarily on the use of hypothetical reasoning during crime investigations, the emphasis in this paper will be on hypothesis generation in the ensuing court room setting. The reason for this shift in emphasis is that our proposal is based on the "abductive diagnosis" paradigm for physical systems diagnosis [1]. Physical systems differ from crime investigation in that most, if not all, component objects of a physical system are known. Typical examples of unknown components in physical systems include leaks in a hydraulic system or short-circuits in an electrical systems. Existing abductive diagnosers can model the behaviour of such unknown components quite easily because they influence the overall physical system at only one location in its topology. Conversely, unknown persons and objects constitute an important feature in crime scenarios and their existence can affect many aspects of the crime scenario: When a body is first found under suspicious circumstances,

[1] Both authors have been supported by EPSRC grant GR/S63267.

the possible causes for the death are considerably broader than the range of possibilities that caused the breakdown of a piece of machinery. Some of the explanations will be "surprising" in the sense that they were impossible to anticipate.

This paper shows how existing diagnostic systems can be amended to incorporate this type of reasoning that is typical for legal proceedings. This paper therefore aims to demonstrate that existing abductive reasoning techniques can not adequately deal with this aspect of reasoning about crime scenarios and to propose a basic extension of our ATMS based decision support system to tackle this restriction. The remainder of this paper is organised as follows: Section 2 presents an overview of the existing decision support system that this paper is based on, Section 3 characterises the class of crime scenarios that are difficult to synthesise with the present approach, Section 4 introduces a novel assumption based peg unification technique that addresses the latter issue, and Section 5 concludes the paper.

2. Background

Although there is no consensus within the wider community of crime investigators as to what constitutes an effective methodology for evidence discovery, crime investigation and forensic argumentation in court, forensic scientists, statisticians and philosophers increasingly advocate the adoption of an abductive reasoning paradigm [2,7,16]. Decision support systems (DSS) aimed at assisting human investigators in aspects of abductive reasoning are inevitably restricted by the limitations of formal inference mechanisms. To alleviate such deficiencies, the approach discussed in [10] combines abduction in the narrow sense (i.e. inverse modus ponens) with model based reasoning techniques such as assumption based truth maintenance. To render the paper sufficiently self-contained, this section will briefly discuss this background material.

2.1. Assumption Based Truth Maintenance

An assumption based truth maintenance system (ATMS) can assist a problem solver by maintaining how each piece of inferred information depends on presumed information and facts, and how inconsistencies arise. This section summarises the functionality of an ATMS as it is employed in this work. For more details, the reader is referred to the original papers [3].

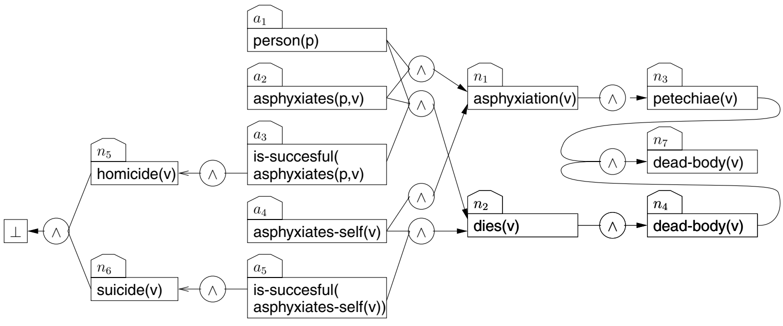

Figure 1. Graphical representation of a sample ATMS.

In an ATMS, each piece of information of relevance to the problem solver is stored as a *node*. Some pieces of information are not known to be true and cannot be inferred easily

from other pieces of information. In the ATMS, such information is represented by a special type of node, called *assumption*. Our model based DSS employs two types of assumptions: (i) conjectures, which cannot be deemed true or false a priori (e.g. "the victim asphyxiated himself"), and (ii) default assumptions, which should be presumed true unless their truth leads to inconsistencies (e.g. "the cause of death proposed by the medical examiner is correct") [15].

Inferences between pieces of information are maintained within the ATMS as inferences between the corresponding nodes. In its extended form, the ATMS can take inferences, called *justifications* of the form $n_i \wedge \ldots \wedge n_j \wedge \neg n_k \wedge \ldots \wedge \neg n_l \rightarrow n_m$, where $n_i, \ldots, n_j, n_k, \ldots, n_l, n_m$ are nodes (and assumptions) representing information that the problem solver is interested in. Figure 1 shows a graphical representation of an ATMS with seven regular nodes $n_1, \ldots, n_7$, five assumption nodes $a_1, \ldots, a_5$, and the following justifications: $a_1 \wedge a_2 \rightarrow n_1$; $a_4 \rightarrow n_1$; $a_1 \wedge a_2 \wedge a_3 \rightarrow n_2$; $a_4 \wedge a_5 \rightarrow n_2$; $n_1 \rightarrow n_3$; $n_2 \rightarrow n_4$; $a_3 \rightarrow n_5$; $a_5 \rightarrow n_6$; and $n_3 \wedge n_4 \rightarrow n_7$. This ATMS provides a very simple description of some of the situations and events that might explain the combined evidence of a dead body of a person v and petechiae on the eyes of v[1].

An ATMS can also take justifications, called *nogoods*, that have lead to an inconsistency. Nogoods are justifications of the form $n_i \wedge \ldots \wedge n_j \wedge \neg n_k \wedge \ldots \wedge \neg n_l \rightarrow \perp$. As such, they impose constraints upon the combinations of assumptions that constitute valid explanation for known observations. The latter nogood implies that at least one of the statements in $\{n_i, \ldots, n_j, \neg n_k, \ldots, \neg n_l\}$ must be false. In the sample ATMS of Figure 1, there is one nogood $n_5 \wedge n_6 \rightarrow \perp$, which indicates that the death of a victim can not be both homicide and suicide.

An *environment* in an ATMS is a set $\{a_1, \ldots, a_m\}$ of assumptions. Each environment describes the possible worlds in which all its assumptions are true. For each of its nodes, an ATMS maintains a description, called a label, denoting the environments that entail it. Because it would be inefficient to store all environments that entail a particular node, a label $L(n)$ of a node n is a set of environments, such that $L(n)$ is *minimal* ($L(n)$ does not contain any supersets of an environment that entails n), *complete* ($L(n)$ contains each environment that entails n or a subset thereof), *sound* ($L(n)$ contains no environment that does not entail n) and *consistent* ($L(n)$ does not contain an environment that entails the nogood node). In the sample ATMS of Figure 1, for instance: $L(n_7) = \{\{a_1, a_2, a_3\}, \{a_4, a_5\}\}$.

Given a set of nodes, assumption nodes, justifications and nogoods, an ATMS can be queried to provide useful information to an abductive problem solver. For example, an ATMS can determine whether a given environment is consistent (i.e. corresponds to a possible world), whether a given environment entails a given node, and which (consistent) environments are sufficient to explain a given node.

2.2. A Model Based Decision Support System for Reasoning about Crime Scenarios

The model based decision support system presented in [10] synthesises an ATMS representing a *space of plausible scenarios* from a given set of available evidence, a given set of facts and a knowledge base of causal rules and constraints upon the consistency of scenarios. The approach follows roughly the following steps:

1. A new ATMS is initialised with one node for each piece of evidence and each fact. Each fact node is justified by the empty environment (in other words, it is deemed true in all possible worlds).

[1]For a substantially extended version of this example, the reader is referred to [10]

2. Through the application of inverse modus ponens of the causal rules in the knowledge base, the ATMS is filled with plausible explanations of the available evidence. In this step, the plausible scenarios that may have caused the evidence are built up and composed within the ATMS.
3. Through the application of modus ponens of the causal rules, the ATMS is filled with plausible consequences of the causes generated in Step 2. In this step, collectible evidence and the assumptions that such evidence depends upon is added to the ATMS.
4. Finally, the constraints of the knowledge base are instantiated in the ATMS in the form of nogoods.

Once this scenario space has been constructed, it can be analysed to provide decision support information by using standard ATMS operations to provide answers to a range of queries, including the following ones:

- Which scenarios can explain the evidence? Scenarios explaining available evidence can be identified by computing the complete and sound set of minimal and consistent environments in the ATMS that entail the available evidence, and then tracing the justifications from the assumptions in the environments and the facts in the ATMS to the given evidence. Our DSS provides the means to describe visually or textually such scenarios.
- Is a given hypothesis about the crime scenario underlying the evidence supported by the available evidence? Any hypothesis is supported by the evidence provided a consistent environment exists in the ATMS that entails both the hypothesis and the evidence.
- What pieces of evidence may be expected to be observed if a certain scenario/hypothesis were true? These are discovered by searching for evidence that logically follows from the/an environment entailing the given scenario/hypothesis, possibly extended with default assumptions.

3. Abductive Inference Mechanisms for Abductive Inference

While the DSS described in the previous section can generate a wide range of plausible scenario to explain a given set evidence, there are certain types of plausible scenarios it is not well suited to synthesise. This drawback is not due to any limitation of abductive reasoning, which is accepted as an appropriate methodology to infer information in reasoning about crime scenarios [16]. Instead, it is due to the limitation of our abductive reasoning algorithm.

The term "abduction" is used in philosophy and artificial intelligence to refer to relatively distinct concepts [17]. In philosophy, "abduction" is associated with the formulation of hypotheses or explanations[2]. In what follows, this concept is named *abductive inference*. In artificial intelligence (AI), "abduction" characterises a specific class of algorithms that implement inference mechanisms enabling knowledge based systems to compute hypotheses or explanations. Hence, the AI concept of "abduction" refers to *abductive inference mechanisms*. While abductive inference mechanisms aim to enable a machine to perform abductive inference, their repertoire of reasoning tends to be substantially narrower than that discussed in philosophy textbooks on abduction. This stems primarily from the fact that computer algorithms need to be substantially more precise than typical analyses of human reasoning.

An example may help illustrate these issues. Assume the body of a man is found in his home. The cause of death is identified as a single shot in the head and blood splatter evidence indicates the man was shot with a small calibre handgun at very short range. No hand gun has

[2]Some authors also include the evaluation of hypotheses, but this topic is beyond the scope of this paper.

been retrieved from the scene. Two witnesses claim to have information that may be relevant to the case. The first heard a male voice shouting threats followed by a single shot. The second saw a suspicious looking person running away from the home of the victim.

Abductive inference may yield several hypothetical causes that explain the pieces of evidence. The victim may have been shot by another person, he may have committed suicide or his death may have been an accident. The absence of the weapon that killed the victim at the crime scene suggests that it was either taken away by a person (such as the victim's killer in the homicide scenario) or that it fell in an awkward location at the crime scene (after a possible suicidal or accidental shooting). The male voice heard by the first witness may have been that of a, say, paranoid delusional victim, or another person, say the killer, threatening the victim. Finally, the suspicious person fleeing the scene may be the victim's killer or a person unrelated to the killing (e.g. the thief of the victim's gun).

The distinct plausible causes for the individual pieces of evidence, which have been identified through causal reasoning, do not yet constitute hypothetical scenarios. These can be identified by composing the plausible causes into coherent combinations of causes that explain all the available evidence. Two (of many) such scenarios in the ongoing example are:

- *Scenario 1*: The perpetrator threatened the victim at the victim's home, killed him, and ran away with the murder weapon, and
- *Scenario 2*: In a paranoid delusion, the victim shouted threats to kill himself and then shot himself in the head; after hearing this, a second person entered the victim's home, saw the dead body, took the weapon, and fled the scene.

Note that this abductive inference involves causal reasoning to identify the situations and events that may have generated the pieces of evidence, and a process that combines the situations and events into coherent hypothetical scenarios. Pierce referred to the latter process as "colligation" and deemed it an inherent feature of abductive (and inductive) inference [14]. Abductive inference mechanisms, however, tend to focus on the causal reasoning task and provide little means for colligation. The ATMS based technique discussed in Section 2 can infer a complete set of minimal and consistent combinations of the causes of the available set of evidence (i.e. solutions to Konolige's model of abduction [12]), but it is not well suited to discern the important possible links between the causes involving unidentified entities.

The latter issue can be explained more precisely by means of the ongoing example. Consider the following three causal rules, which may be part of a knowledge base that aims to aid in the synthesis of plausible scenarios, such as Scenario 1:

$$\begin{gathered}\text{person}(P) \wedge \text{victim}(V) \wedge \text{scene}(S) \wedge \text{at}(P,S) \wedge \text{near}(W,S) \wedge \\ \text{threatened}(P,V) \wedge \text{near}(W,P) \rightarrow \text{witness}(W, \text{threat}(\text{near}(S))\end{gathered} \tag{1}$$

$$\begin{aligned}\text{person}(P) \wedge \text{scene}(L) \wedge \text{shot}(S,P,V,L) \wedge \text{gun}(S,G) \wedge \text{range}(S,R,) \rightarrow \\ \text{evidence}(\text{shot-at-range}(V,R)) \wedge \text{evidence}(\text{shot-with}(V,G))\end{aligned} \tag{2}$$

$$\text{person}(P) \wedge \text{scene}(S) \wedge \text{took}(P,G) \rightarrow \neg\text{evidence}(\text{recover}(G,S)) \tag{3}$$

Rules (1), (2) and (3) respectively provide explanations for the testimony of the first witness, the shot at short range with a handgun, and the absence of the weapon that killed the victim at the crime scene. In each of these rules, P refers to *an* unknown person. Therefore, P is considered to be existentially quantified.

While these rules can be employed by the abductive inference mechanism of Section 2 to produce useful explanations, the semantics of first order predicate logic can not distinguish

between subtle but important variations of Scenario 1. The closest scenario that our abductive inference mechanism can produce to Scenario 1 is one where three different instances of P refer to the person threatening the victim, the person killing the victim and the person taking the gun from the scene. The very plausible case that the three persons are one and the same is not differentiated from its alternatives, even though this is a crucial distinction.

Situations such as the one in the example arise frequently in the crime investigation and legal reasoning domains. Because evidence usually relates to aspects or features of persons and objects, and the events and situations in which they occur, it is not always possible to uniquely identify these entities from the start of the investigation, or even during the subsequent court proceedings (as argued in the introduction). Nevertheless, the possible number of unidentified entities relevant to the crime must be considered and taken into account. The remainder of this paper proposes an approach to extend our abductive reasoning mechanism accordingly, thereby expanding the applicability of abductive inference mechanisms.

4. Assumption Based Peg Unification

The task of identifying different references to the same entity is known as *coreference resolution* in computational linguistics. In the analysis of a discourse, it is important that references to the same entity are correctly associated with one another because each of the expressions that contains one of these references may add some information about the entity in question. For example, in the sentence "Every farmer who owns *a donkey*, beats *it*." "a donkey" and "it" refer to the same entity. The first half of the sentence conveys that the entities of interest are all donkeys owned by farmers. The second half of the sentence communicates that the entities of interest are beaten. Thus, the sentence as a whole imparts the knowledge that all donkeys owned by farmers are beaten.

A wide range of techniques has been devised to perform coreference resolution tasks, such as the one illustrated in the example. The vast majority of these techniques specialise in examining texts for discourse analysis or information extraction. An important property of the existing approaches is that they tend to consider only a single possible solution at any one time, while the present problem domain requires a method that can represent and reason with multiple possible worlds simultaneously.

4.1. Pegs

The objective of this work is to identify possible references to the same unknown or partially specified entities in the scenario space. In order to correctly distinguish such entities, the notion of *pegs* is adopted from the literature on coreference resolution [8,13]. Pegs refer to a specific entity whose exact identity remains unknown (or partially specified). In this paper, each peg is identified by an expression of the form $\_n$, where n is a non-negative natural number. At the start of the scenario space generation algorithm $n = 0$, and n is incremented by 1 after each generation of a new peg. As such, each new peg is identified uniquely.

New pegs may be introduced into the scenario space during the instantiation of causal rules of the form `if` $\{A_n\}$ `assuming` $\{A_s\}$ `then` $\{c\}$, where A_n is a set of antecedent predicates, A_s is a set of assumption predicates and c is a consequent predicate. Whenever a rule, whose antecedent or assumption predicates contain variables that do not occur in the consequent sentence, is applied during the inverse modus ponens phase of the scenario space generation algorithm (i.e. step 2), then those variables are instantiated by pegs. Consider, for instance, applying inverse modus ponens on rule

`if` $\{\text{scene}(S)\}$ `assuming` $\{\text{person}(P), \text{took}(P,G)\}$ `then` $\{\neg\text{evidence}(\text{recover}(G,S))\}$ given the piece of evidence: $\neg\text{evidence}(\text{recover}(\text{handgun}, \text{home}(\text{victim}))\}$. The required substitution $\{G/\text{handgun}, S/\text{home}(\text{victim})\}$ does not provide an instance for P. Here, P refers to an unknown entity and it is therefore substituted by a peg, say, _0. Therefore, the assumptions $\text{person}(\_0)$ and $\text{took}(\_0, \text{handgun})$ are added to the scenario space.

Similarly, pegs may also be introduced during the modus ponens phase of the scenario generation algorithm (i.e. step 3). In this case pegs are introduced when a rule whose consequent predicates contain variables that do not occur in the antecedent or assumption sentences, is applied.

4.2. Peg Unification

Because a peg refers to an unknown entity, it can be treated as a constant that uniquely identifies a that entity, or it can be unified with a ground term, including another peg or terms containing other pegs. In the latter case, the unification is possible if it is hypothesised that the entity represented by the peg and the entity represented by the term unified to the peg are the same one. This hypothesis must therefore be made explicit by means of an assumption whenever an inference is made that depends on the unification of a peg and a ground term. In the remainder of this paper, such assumptions are referred to as *peg unification assumptions*.

In this paper, each peg unification assumption takes the form $\text{bind}(\_n, t)$, where $\_n$ is a peg and t is a ground term (which may include a peg). A peg unification assumption $\text{bind}(\_n, t)$ is added to the scenario space for each pair of predicates that can be matched using a substitution that contains a mapping of the form $\_n/t$.

The binding relation implied by these assumptions is transitive. Therefore, peg unification can not only be assumed, but also be entailed by other peg unification assumptions. This knowledge is represented explicitly in the scenario space: for each pair of peg unification assumptions $\text{bind}(\_i, t_1(\ldots, \_j, \ldots))$ and $\text{bind}(\_j, t_2(\ldots, \_k, \ldots))$, the following new justification is added to the emerging scenario space:

$$\text{bind}(\_i, t_1(\ldots, \_j, \ldots)) \wedge \text{bind}(\_j, t_2(\ldots, \_k, \ldots)) \rightarrow \text{bind}(\_i, t_1(\ldots, t_2(\ldots, \_k, \ldots), \ldots))$$

4.3. Scenario Space Generation

Peg unification affects the way the scenario space generation algorithm operates, when applying causal rules of the form shown in 4.1 in steps 2 and 3 of the algorithm and constraints of the form `inconsistent {` I `}`, where I is a set of sentences that should be deemed nogood, in step 4 of the algorithm. In the extended approach, each application of a causal rule or constraint requires the following operations:

1. *Unify* the relevant sentences (i.e. the consequent of the causal rule during inverse modus ponens, the antecedents of the causal rule during modus ponens, or the inconsistent sentences of the constraint) with nodes in the emerging scenario space, and return the substitution σ required to achieve the unification.
2. *Record* each binding that unifies a peg with a term in the scenario space and a newly created set A_p. That is, for each binding $\_n/t \in \sigma$ or $t/\_n \in \sigma$, where $\_n$ is a peg and t is a term that the $\_n$ is unified with, the peg unification assumption $\text{bind}(\_n, t)$ is added to A_p.

3. *Instantiate* the remaining sentences (i.e. the antecedents and assumptions during inverse modus ponens or the assumptions and consequent during modus ponens) by applying the substitution σ and add those that do not already exist in the scenario space as new nodes.
4. *Generate* a justification of the form $\left[\wedge_{(a \in A_n \cup A_s)} \sigma a\right] \wedge \left[\wedge_{(u \in A_p)} u\right] \rightarrow \sigma c$ in case of an application of a causal rule, or a nogood of the form $\left[\wedge_{(a \in I)} \sigma a\right] \wedge \left[\wedge_{(u \in A_p)} u\right] \rightarrow \perp$ in case of an application of a constraint.

Consider, for instance, the following rule expressing the constraint that a person should be either male or female: $\text{person}(P) \wedge \text{gender}(P, \text{male}) \wedge \text{gender}(P, \text{female}) \rightarrow \perp$ and let the scenario space contain to references to persons, i.e. $\text{person}(\_0)$ and $\text{person}(\_1)$. Naturally, _0 and _1 may refer to the same person. Therefore, the antecedents of the gender constraint can match the ground terms $\text{person}(\_0)$, $\text{gender}(\_0, \text{male})$, and $\text{gender}(\_1, \text{female})$ assuming the peg unification assumption $\text{bind}(\_0, \_1)$ is made. Therefore, a valid instantiation of the constraint is $\text{person}(\_0) \wedge \text{gender}(\_0, \text{male}) \wedge \text{gender}(\_1, \text{female}) \wedge \text{bind}(\_0, \_1) \rightarrow \perp$.

The last example has show how additional predicates can describe certain features of the entities represented by the pegs and how such predicates may impose constraints on the plausibility of peg unification assumptions. It is important to point out that these features attributed to unknown entities, as with other information stored in an ATMS, can be modelled as dependent upon conjecture. As such, the approach allows for different pieces of evidence regarding similar entities to be modelled to refer to: (i) correctly interpreted evidence of multiple entities, (ii) conflicting evidence regarding the same entity, or (iii) evidence that bears no relation to any entity in existence.

4.4. Scenario Extraction

In [10], we argue that a scenario driven decision support system (DSS) for crime investigation should explain the available evidence by means of the simplest scenario that entail this evidence and one of the main hypotheses of interest. For example, in the investigation of a suspicious death, our DSS provides the simplest scenarios that explain homicide, suicide, accidental death and natural causes. In [10], the simplest scenarios are defined as those entailed by the environments with the smallest number of conjectures. However, all other things being equal, scenarios with fewer entities are normally deemed to be less complex. Therefore, peg unification assumption simplify scenarios, and for that reason they treated as default assumptions.

While the ATMS label propagation algorithm provides an efficient means to determine the minimal (or smallest) environments (of conjectures) that entail a given set of nodes (representing pieces of evidence), it does not provide a facility to determine the maximum number of default assumptions. However, the basic candidate generation algorithm of the General Diagnostic Engine (GDE) [5] can be employed to extend an environment E of conjectures with a consistent set of peg unification assumptions. The candidate generation algorithm can be applied as follows:

1. Let P be the set of all pegs that occur in predicates that logically follow from E. Let E_p be the set of all peg unification assumptions that contain pegs in P and no other pegs. In other words E_p is the set of all peg unification assumptions the E can be extended with.
2. Let N be the set of nogood environments that are subsets of $E_p \cup E$. Formally, $N = \{E_i \mid E_i \subseteq E_p \cup E, E_i \in L(\perp)\}$. Each environment in N is a combination of peg unification assumptions that is inconsistent. Therefore, one peg unification as-

sumption from each environment in N needs to be removed from E_p in order to make it consistent.

3. If N is empty, then return the singleton $L_E = \{E \cup E_p\}$ and end the algorithm. Otherwise, proceed to step 4.
4. Let N' be the set of all combinations of peg unification assumptions from E_p that are inconsistent with E. In other words, N' is the set constructed by removing all conjectures in E from each of the nogood environments in N. Or formally $N' = \{E'_i \mid E'_i = E_i - E_p, E_i \in N\}$
5. Generate the set C of GDE candidates from the nogood environments in N' as follows: (i) Assemble a set C of all the cross product sets of the nogood environments in N. Each cross product set in is a set that contains at least one assumption from each of the environments in N. (ii) Remove each cross product set from C that is a superset of another cross product set in C. Each candidate set in C is a minimal set of peg unification assumptions from E_p that is not consistent with E.
6. Let $L_E = \{E' \mid E' = E \cup E_p - E_c, E_c \in C\}$. In other words L_E is set of all supersets of E that contain all the peg unification assumptions in E_p minus a candidate set from C.

The set L_E returned by the above procedure contains all consistent extensions of E with relevant peg unification assumptions. To illustrate this approach, consider the graphical representation of a partial scenario space in Figure 2.

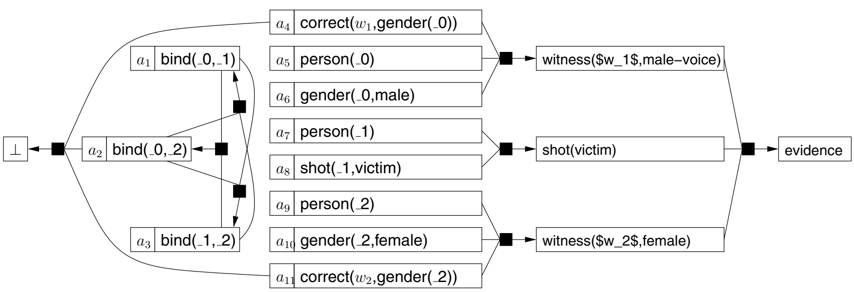

Figure 2. Sample scenario.

In Figure 2, the node evidence, which represents all the available evidence, is supported by a single environment of conjectures $E = \{a_4, \ldots, a_{11}\}$. This environment can be extended with peg unification assumptions as follows: Step 1: The set of relevant pegs $P = \{\_0, \_1\}$ and hence, the set of relevant peg unification assumptions $E_p = \{\text{bind}(\_0, \_1), \text{bind}(\_1, \_2), \text{bind}(\_0, \_2)\}$; Step 2: The set of nogood environments $N = \{\{a_1, a_3, a_4, a_{11}\}, \{a_2, a_4, a_{11}\}\}$; Step 4: The combinations of peg unification assumptions inconsistent with E is $N' = \{\{a_1, a_3\}, \{a_2\}\}$; Step 5: The minimal candidates that can be generated from N' is $C = \{\{a_1, a_2\}, \{a_2, a_3\}\}$; Step 6: The set of valid extensions of E is $L_E = \{\{a_3\}, \{a_1\}\}$. Thus, the environments entailing the available evidence with minimal conjectures and maximal peg unification is: $\{\{a_3, a_4, \ldots, a_{11}\}, \{a_1, a_4, \ldots, a_{11}\}\}$.

5. Conclusions and Future Work

Building on the preliminary work presented in [9], this paper has characterised further the benefits of peg unification in abductive reasoning about crime scenarios and basic approach has

been proposed. The novelty and research contribution of this approach is twofold. Firstly, the new representation formalism and inference mechanisms bring the benefits of peg unification from the computational linguistics and information extraction domains to the entirely new domains of reasoning about crime scenarios and evidence evaluation. As such, this work can enable abductive inference mechanisms in the latter domains to accomplish a broader range of abductive inference. Secondly, the peg unification approach is the first one to be integrated into an assumption based truth maintenance system (ATMS). While conventional peg unification mechanisms aim to find a single most likely set of coreferences, the use of the ATMS in peg unification enables the associated problem solver to reason about multiple plausible sets of coreferences that may be valid in different possible worlds.

As the approach presented herein is an initial proposal to incorporate peg unification into abductive synthesis of crime scenarios, a number of important issues remain to be addressed in future work. While the approach presented herein facilitates modelling certain types of abductive reasoning about crime scenarios, it does not address the broader problem of knowledge acquisition for this type of reasoning. Future work intends to tackle this issue by developing techniques aimed at extracting cause-and-effect knowledge from individual cases as they are encountered. Also, the important issue of time and space complexity of the approach proposed herein needs to be studied. Because ATMSs are constraint propagation mechanisms, they eliminate the need for search algorithms, such as backtracking and its variants. But that need not be the most efficient approach to identify consistent crime scenarios, and hence, the use of constraint satisfaction algorithms should be examined. Other future work will examine the integration of the peg unification approach into our Bayesian scenario space synthesis approach [11] and the enrichment of the existing knowledge representation formalism.

References

[1] L. Console and P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 7(3):133–141, 1991.

[2] R. Cook, I. Evett, G. Jackson, P. Jones, and J. Lambert. A model for case assessment and interpretation. *Science and Justice*, 38:151–156, 1998.

[3] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[4] J. de Kleer. A comparison of ATMS and CSP techniques. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 290–296, 1989.

[5] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[6] D. Dixon. Police investigative procedures. In C. Walker and K. Starmer, editors, *Miscarriage of Justice. A Review of Justice in Error*, 1999.

[7] A. Jamieson. A rational approach to the principles and practice of crime scene investigation: I. principles. *Science & Justice*, 44(1):3–7, 2004.

[8] L. Karttunen. Discourse referents. *Syntax and Semantics*, 7:363–385, 1976.

[9] J. Keppens and B. Schafer. "murdered by persons unknown" - speculative reasoning in law and logic. In *Proceedings of the 17th Annual Conference on Legal Knowledge and Information Systems (JURIX)*, pages 41–51, 2004.

[10] J. Keppens and B. Schafer. Knowledge based crime scenario modelling. *Expert Systems With Applications*, 30(2), 2006.

[11] J. Keppens, Q. Shen, and B. Schafer. Probabilistic abductive computation of evidence collection strategies in crime investigation. In *Proceedings of the 10th Internation Conference on Artificial Intelligence and Law*, 2005.

[12] K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 52:255–272, 1991.

[13] F. Landman. *Towards a theory of information. The status of partial objects in semantics*. Foris, Dordrecht, 1986.

[14] C. Peirce. *Collected Papers*. Harvard University Press, 1933. Edited by C. Hartshorne and P. Weiss.

[15] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[16] D. Schum. *The Evidential Foundations of Probabilistic Reasoning*. Northwestern University Press, 1994.

[17] P. Thagard and C. Shelley. Abductive reasoning: Logic, visual thinking, and coherence. In L. Dalla Chiara et al., editor, *Logic and scientific methods*, pages 413–427. Kluwer, 1997.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Dynamics of Rule Revision and Strategy Revision in Legislative Games

Moshe Looks [a,1], Ronald P. Loui [a] and Barry Z. Cynamon [b]
[a] *Dept. of Computer Science, Washington University in St. Louis, USA*
[b] *Graduate School of Business, University of Chicago, USA*

Abstract. Many legislative games of interest defy classical assumptions and techniques; they tend to be open-ended, with weakly defined objectives, and either non-competitive or pseudo-competitive. We introduce a conceptual and mathematical framework for grappling with such systems. Simulation results are presented for basic specifications of the framework that exhibit a number of qualitative phenomena overlapping with real-world dynamics across a broad spectrum of settings, including aspects of financial regulation and academic decision procedures, that as we demonstrate, may be viewed through the lens of our framework.

Keywords. Multiagent Systems, Social Simulation, Legislative Revision

1. Introduction and Overview

It is customary for mathematical economics and political economy to assume that self-interested social agents can be effectively incentivized in the presence of the right economic controls. Almost every institution seeks to achieve its goals by publishing criteria for rewarding and penalizing the behavior of agents. In this way, legislators seek a meritocratic society that provides more room for free choice than an authoritarian society.

There are many problems with the assumption of effective incentivization. For one thing, agents may place value on unanticipated substantive dimensions, unforseen by the legislators. Worse, agents are known to attach value to procedural, process, and contextual components of their experience, such as the valuing of individuality, the disvaluing of conformity or compliance, the mechanization of their behavior, or the fairness of the outcomes, whether the outcomes result mainly from individual contribution to collective action or not. In many cases, the non-ideal incentive drives the behavior of the system more than the non-ideal behavior of the agents, or the non-ideal responsiveness of the process.

A couple of issues are appropriate for investigation through agent modeling and computational simulation of the resultant society. Here, we are concerned with the legislator's problem of *abridgement*, which results from the legislator's inability to publish incentives that are precisely in line with the social objectives. In jurisprudence there is a tradition of assuming that the rules are imperfect mainly for linguistic reasons, the legis-

[1]Correspondence to: Moshe Looks, Dept. of Computer Science, Washington University, St. Louis, MO 63130, USA Tel.: +1 202 641 2157; mlooks@cse.wustl.edu. Also loui@cse.wustl.edu.

lature's inability to express explicitly and precisely the behavior that is desired. There are also pragmatic reasons why abridgement occurs: the process of producing controls has many imperfections, such as a limit of agreement among conflicting legislative views, a limit to revision, a limit to the practicable complexity of rules, and a limit to the legislators' understanding of the agents it seeks to incentivize.

Most importantly, we consider the race conditions that occur when the legislature has bounds on how quickly it can revise its edicts, and the agents have bounds on how quickly they can revise their positions.

All of these issues are within the purview of the proposed model. In addition, in a nod to Herbert Simon, agents are restricted in their knowledge of admissible strategies, both in terms of their legality and their potential as a revised position.

The aim of this paper is to produce a simple and flexible mathematical model that captures some of the interesting phenomena we claim to observe in practice. In this way, we seek a better understanding of the process of controlling agents through the revision of non-ideal incentives.

2. Related Work

Broadly speaking this paper is founded on the use of multi-agent system models within computational social science, as advocated/demonstrated in [7,9,5]. There is also prior work focusing in particular on incentivizing agents; e.g., with respect to violation of norms (*cf.* [4,3]), and specifi c scenarios such as rent control [2] and water demand [11]. More generally, our work is based on fundamental concepts of bounded rationality [13], legal positivism [10], and the study of incentives for individual advancement [14].

3. Veridical Utility and Public Incentive

Consider a legislature incentivizing a population of agents toward some goals. Real goals are complex and may be implicit; they must be approximated via a legislative *abridgement*, $\mathbb{A}$. The global utility function or *truth* ($\mathbb{V}$), embodying the actual goals, is over some very high dimensional space where agents live.[1] Over time, agents move through this space by changing their *positions*. A position corresponds to a strategy, action, or mode of behavior in the given domain (e.g., a particular kind of stock sale, managerial style, research focus, etc.).

Assume a veridical utility function mapping from $\mathbb{R}^n$ to $\mathbb{R}$; the legislative abridgement used for public incentive, as a function of agent position $\vec{x} \in \mathbb{R}^n$, might be

$$\mathbb{A}(\vec{x}) = \mathbb{V}(\vec{x}A), \qquad A \in \mathbb{R}^{n,n}, \qquad rank(A) = k \ll n. \tag{1}$$

Here, $\mathbb{A}$ downsamples $\mathbb{V}$ by projecting n-dimensional points onto a k-dimensional hyperplane passing through the n-dimensional space. In contrast to $\mathbb{V}$, $\mathbb{A}$ is assumed to be publicly available, explicit, and relatively inexpensive to compute.

[1] Throughout this paper, we assume that $\mathbb{V}$ may be additively summed over agents; this need not be the case, but is often a useful simplification.

The social counterparts of veridical and abridged utilities are simply the summations over the entire population; we will assume a population of N agents, with positions $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N$. Agents are not omniscient. Each agent has a set of known positions, $\Upsilon_i \subseteq \mathbb{R}^n$, to choose from. Furthermore, many positions are legislatively inadmissible, represented by a set of banned positions, $\Omega \subset \mathbb{R}^n$, which is determined by the legislature.

4. Change

At discrete time steps, all agents may update their positions. Similarly, the legislature may update $\mathbb{A}$ and Ω. This may be denoted by subscripting them with a time-step (e.g., $\mathbb{A}_t$ and $\vec{x}_{i,t}$ specify the legislative abridgement and ith agent's position at time t, respectively). The spaces over which these are defi ned are assumed to be static, however.

If $\mathbb{A}$ perfectly mirrors $\mathbb{V}$, the legislature has no need for rule revision. When this is not the case, inaccuracies in the abridgement can lead to *legislative misdirection*. That is, agents end up in the wrong regions of the space (from a $\mathbb{V}$-maximizing perspective), and the public incentives must be revised to compensate.

4.1. Bounds

There are often temporal constraints on the rates of legislative *change* and *growth* over time. Legislation is ρ-δ-bounded if

$$\begin{array}{ll} \forall t{\geq}0, \quad |A_t, A_{t+1}| \leq \rho & \forall t{\geq}0, \quad |\Omega_t, \Omega_{t+1}| \leq \rho \\ \forall t{\geq}0, |A_{t+1}| - |A_t| \leq \delta & \forall t{\geq}0, |\Omega_{t+1}| - |\Omega_t| \leq \delta, \end{array} \tag{2}$$

where $|\cdot,\cdot|$ and $|\cdot|$ are normalized metrics of distance and size (or complexity), respectively, over $\mathbb{R}^n \to \mathbb{R}$ on the left, and $2^{\mathbb{R}^n}$ on the right.

Constraints on an agent's change (in position) and growth (in the set of known positions) are analogous. The ith agent ($1{\leq}i{\leq}N$) is λ-μ-bounded if

$$\begin{array}{ll} \forall t{\geq}0, \quad |\vec{x}_{i,t}, \vec{x}_{i,t+1}| \leq \lambda & \forall t{\geq}0, \quad |\Upsilon_{i,t}, \Upsilon_{i,t+1}| \leq \lambda \\ \forall t{\geq}0, |\vec{x}_{i,t+1}| - |\vec{x}_{i,t}| \leq \mu & \forall t{\geq}0, |\Upsilon_{i,t+1}| - |\Upsilon_{i,t}| \leq \mu, \end{array} \tag{3}$$

with normalized metrics $|\cdot,\cdot|$ and $|\cdot|$ over $\mathbb{R}^n$ (left) and $2^{\mathbb{R}^n}$ (right). A population of agents may be said to be λ-μ-bounded if all of its member are.[2]

5. Dynamics

In this section, we will outline in more detail various agent and legislative strategies which can give rise to the emergent dynamics we are interested in studying.

[2] Note that if an agent's position is *extinguished* (i.e., added to Ω) λ-μ-bounds may need to be violated to move to a new position. Similarly, if $\Upsilon_{i,t} \cap \Omega_t = \emptyset$, an agent may be unable to field *any* position.

5.1. The Character of Simple Agents

In the simplest case, all agents are interchangeable, but begin play with different initial positions. The set of known positions may simply be a spherical neighborhood of the player's position defined by a constant radius (d). More powerful agents might be specified by increasing d. At any given time-step, agents simply pick whichever known position maximizes the legislative abridgement (i.e., hill-climbing). Putting it all together, we have

$$\begin{aligned} &\forall 1 \le i \le N, && \vec{x}_{i,0} = c_i \\ & && \Upsilon_{i,0} = \{\vec{x} \in \mathbb{R}^n : |\vec{x}, \vec{x}_{i,0}| \le d\} \\ &\forall 1 \le i \le N, t > 0, && \vec{x}_{i,t} = \arg\max_{\vec{x} \in \Upsilon_{i,t-1}} A_t(\vec{x}) \\ & && \Upsilon_{i,t} = \{\vec{x} \in \mathbb{R}^n : |\vec{x}, \vec{x}_{i,t-1}| \le d\}, \end{aligned} \tag{4}$$

where c_i, the initial agent locations, are drawn from some prior distribution. Such agent behavior is clearly λ-μ-bounded, with $\lambda = d, \mu = 0$, using straightforward metrics (in this context, behavioral complexity is assumed to be constant). This protocol for agent behavior (or, *agent strategy*) will be refereed to as $greedy_{A,d}$.[3]

However, many of the effects we are interested in studying will only emerge from a substrate containing inter-agent effects. The simplest of these is *imitation*. Let $\overline{\vec{x}}_t$ be the average of all agent positions at time t. We may define a new agent strategy, $imitate_{A,d}$, as moving distance d from one's previous position toward $\overline{\vec{x}}_t$. This behavior is also clearly λ-μ-bounded ($\lambda = d, \mu = 0$). We may also of course consider composite agents that weight greediness versus imitation, or alternate between greedy and imitative steps.

5.2. Simple Agents with Idealized Public Incentive

Before proceeding to more complex agents and dynamic legislation, let us demonstrate a few basic effects with the simple agents described above and idealized public incentive (hence a static legislature). Here we will have an unchanging $\mathbb{V} = \mathbb{A}$, reducing the model to a swarm of optimizers with a fixed utility function. Consider $n = 2$ and the extremely simple one-peak function $A(s) = -|\vec{x}, (0,0)|$, where $|\cdot,\cdot|$ is normalized Euclidean distance.[4]

The initial player positions are uniformly randomly sampled from $[0,1] \times [0,1]$. Position selection is implemented by sampling from the current position, along with 100 uniformly spaced points on a circle with radius $d = 0.01$ centered at the current position. Simulations are run for 100 time-steps, with 20 agents.

Figure 1 shows the results for three different agent populations. The first (upper-left) consists entirely of greedy agents that converge quickly to the global optimum. The second (upper-right) consists of imitative agents that fail to reach the global optimum. Since none of the agents are novelty-seekers, the global utility will remain roughly con-

[3] As a technical note, agents here are exploring what might be called the $\mathbb{V}$-neighborhood of their positions, rather than the $\mathbb{A}$-neighborhood. To clarify, $\mathbb{A}$ will typically map agent position to a subspace (i.e., reducing the intrinsic dimensionality of the space from n to k). However, this does not circumscribe agent explorations; they search through the full n-dimensional space, albeit in a very incremental and limited fashion.

[4] It is important to note that we are not interested in function optimization per se; rather, these examples are used to illustrate the emergence of interesting dynamics in our model that mirror some real-world phenomena.

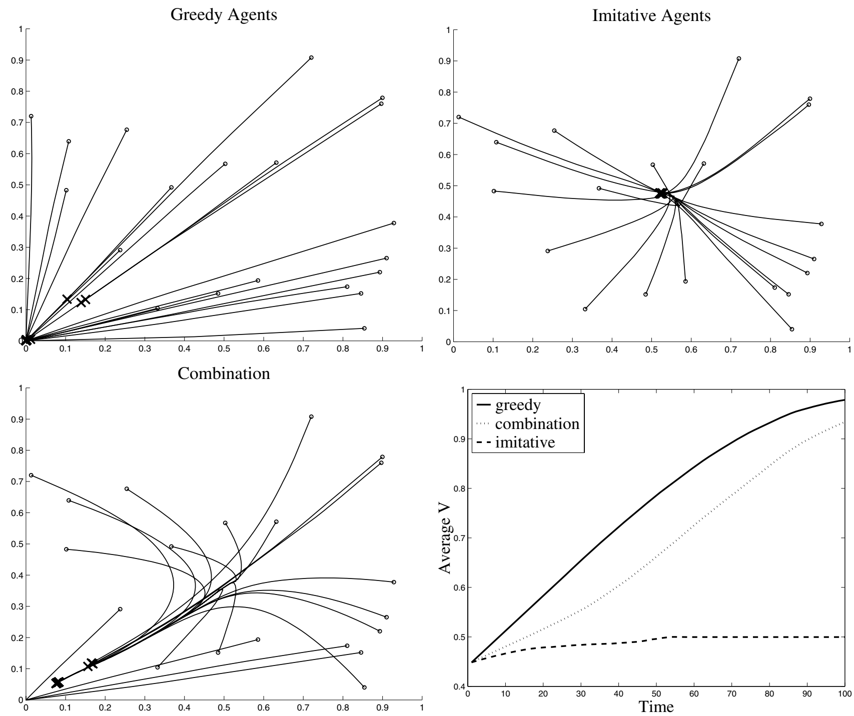

Figure 1. Trajectories for three populations of agents over time: greedy, imitative, and a combination of the two. The global optimum is the lower-left corner, $(0, 0)$. Initial agent positions are the same for all populations, and indicated by small dots. Final agents positions are denoted by $\times$es.

stant (see lower-right). The third case (lower-left) consists primarily of imitative agents, combined with a small number (4) of greedy agents that act as bellwethers for the population as a whole. Since these agents are imitated, but do not imitate, they will have a disproportionate effect on the fi nal outcome of the simulation.

5.3. *The Character of Simple Legislatures*

In the simplest case, the legislature is merely reactive, modifying A (used to compute $\mathbb{A}$ as described above) and Ω in response to trends in agent movement through the space. The set of inadmissible positions may be implemented as an initially empty queue that is slowly fi lled with agent positions which are undesirable from the legislature's perspective as a $\mathbb{V}$-maximizer. If the legislature is unable to compute this directly, it may be approximated via a background function (described below). Similarly, $\mathbb{A}$ may be adjusted (e.g., in terms of which dimensions are supported) to maximize the expected gain in $\mathbb{V}$ with respect to current agent positions.

This legislature will be ρ-δ-bounded, with $\delta = 0$. The exactly value of ρ will depend on the particular updating rule and distance metric used. For example, if at most m dimensions are modifi ed, and the distance between legislations is the number of dimensions by which they differ, we will get $\rho = m$.

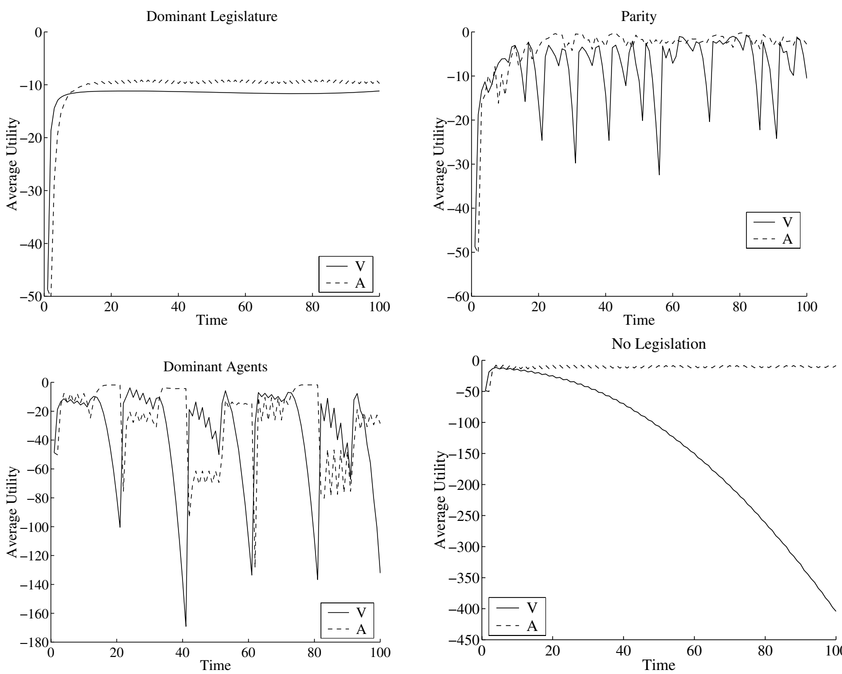

Figure 2. Veridical and abridged utility over 100 time steps for four legislative configurations.

5.4. Simple Agents with Circumscribed Public Incentive

This section will explore simulations where public incentive is circumscribed, leading to an abridgement that changes over time. Again we will consider $n = 2$, but this time with $k = 1$. The legislator is limited to mapping the agents' positions onto a line. The function to be optimized here is again quite simple, $-(1-x)^2 - (1-y)^2$, which has a global optimum at $(1, 1)$. Initial agent positions are uniformly drawn from $[0, 10] \times [0, 10]$.

At each *legislative update*, a new $\mathbb{A}$ may be computed (Ω is not used here). That is, the legislature must choose a line to project the agent positions onto. Ideally, each agent would be mapped onto a line passing through its own position and the global optimum. Since the legislation must be identical for all agents however, this is generally not possible. Instead, a line passing through the optimum and the average of all agent positions, $\overline{\vec{x}}_t$, is used.

What dynamics will arise if all of the agents follow the $greedy_{A,d}$ strategy? It depends on how often legislative updates (i.e., rule revisions), occur. Figure 2 shows average $\mathbb{V}$ and $\mathbb{A}$ values over 100 time steps for four regimes; dominant legislature (updates at every time step), parity (updates every other time step), dominant agents (updates every ten time steps), and no legislative updates.

What is going on here? Let's first consider the case where there are no legislative updates (lower-right). To begin with, agents are directed toward the optimum, and both $\mathbb{V}$ and $\mathbb{A}$ rise. Quickly however the legislation becomes out of date, and while $\mathbb{A}$ remains roughly constant, it no longer mirrors veridical utility, which falls dramatically as agents move further and further away from the high-$\mathbb{V}$ regions of the space.

Next, consider a dominant legislature (upper-left). Both $\mathbb{A}$ and $\mathbb{V}$ rise precipitously, as expected. But why is the global optimum (0) never reached? The problem is that we are dealing with many agents, and there is no mechanism for instituting personalized legislation. That is, the simulation reaches a local optimum where it is simply not possible for the legislature to nudge any agents closer to the global optimum without nudging other agents correspondingly further away.

Finally, some words on the most dramatic scenarios, dominant agents (lower-left) and parity (upper-right). In the former, legislature has some power to mitigate disastrous crashes of $\mathbb{V}$, but only at the cost of introducing similar chaotic fluctuations in $\mathbb{A}$, as infrequent but massive legislation wracks the fi tness landscape. The latter still experiences periodic crashes of $\mathbb{V}$ as the legislature falls temporarily behind, but at the same time, reaches the highest levels of $\mathbb{V}$ more often than any of the other scenarios shown.

5.5. Agent Behavior in Complex Domains

The agent behavioral schema outlined above neglected the issue of strategy complexity by assuming that all strategies were of equal complexity. To understand how this may not be the case, consider a space $\mathbb{R}^n$ with very large n, where the starting positions of agents are zero along the vast majority of the dimensions. Non-zero dimensions are said to *support* a particular strategy; $support(s) = \{1 \leq i \leq k : \vec{x}[i] > 0\}$, where $\vec{x}[i]$ denotes the ith component of strategy $\vec{x}$. We may now defi ne the complexity of a given position, $|s|$, as $|support(s)|$.

Now, assume a λ-μ-bounded population; movement along known dimensions and discovery along new (previously unsupported) dimensions may both be limited, to differing extents. This leads to a familiar dichotomy of exploration (seeking new dimensions worth optimizing on) versus exploitation (optimizing along currently supported dimensions).

5.6. Legislative Behavior in Complex Domains

Legislation operating in a complex domain may induce diverse and nuanced constraints. For example, an agent may discover a previously unknown strategy (position) along a new dimension which spreads to other agents via imitation and plays havoc with $\mathbb{V}$ (possibly leading to a crash). The legislature may retaliate via modifying $\mathbb{A}$ and/or extending Ω (leading to *strategy extinction*), but is limited here by ρ-δ-bounds.

5.7. Background Functions

To improve the model, consider a *background* function ($\mathbb{B}$), providing additional insights into $\mathbb{V}$, which is not determined by the legislature. Agents that integrate this background knowledge into their choice of strategies may have a better chance of improving $\mathbb{V}$. Furthermore, such agents might be able to avoid strategy extinction, as a region of the space with high $\mathbb{A}$ and low $\mathbb{B}$ would be a prime target for future legislative action.

Background knowledge can also play an important role in overcoming local optima and/or deceptive regions of the search space with respect to $\mathbb{A}$. It may alternatively be seen as injecting new strategies into play. In a population where some agents are imitative, these innovations can then spread to others.

With respect to some background knowledge $\mathbb{B}$, one might taxonomize agents into four classes as follows:

- **positive-novel** agents try to maximize $\mathbb{B}$, without regard for $\mathbb{A}$.
- **nonpositive-novel** agents try to maximize $\mathbb{B}$, while avoiding decreasing $\mathbb{A}$.
- **positive-compliant** agents try to maximize $\mathbb{A}$, while avoiding decreasing $\mathbb{B}$.
- **nonpositive-compliant** agents try to maximize $\mathbb{A}$, without regard for $\mathbb{B}$.

According to this taxonomy, $greedy_A$ agents, for example, would be classified as *nonpositive-compliant.*

6. Scenarios

There is perhaps no better real-world example of the type of legislative games we address in this paper, and hence inspiration for future work, than the complex financial markets that allocate capital in the modern global economy. These scenarios exhibit great complexity in terms the interactions by between regulator and regulated (strategy extinction vs. strategy discovery).

The legislator's motivation for intervening in financial markets is unquestionable. Given populations' exposure to finance, and the tremendous power of the financial institutions that control the flow of capital, the finance industry is seen as being too important to be left to its own devices. Every time there is a bank failure or an investment scandal, the resulting hysteria either gives members of the legislature an opportunity to push through a bill, or forces them to in order to appease their frightened or angry constituents.

Thus the legislator bans certain strategies while encouraging others. It uses rule revision to incentivize such behavior as it expects to reduce vulnerability, over-exposure, and over-borrowing, in order to promote long-term cautious investment. The agents respond with an impressive rate of strategy discovery. As Pixley [12] writes, "the [financial] sector has diced and packaged social relations of debts and mortgages, of future promises into future securities, futures, and derivatives of futures, into reinsuring assurances of insurance."

For example, commercial banks are required by law to hold a certain percentage of all funds deposited in checking accounts aside. This is known as a reserve requirement, and is intended to decrease volatility. Furthermore, it allows the Federal Reserve system to be self-funded, as it accrues more than enough to meet its operating expenses from interest on the treasury bills that it purchases with banks' reserves. Thus, congress has a clear incentive to maintain the reserve requirement—without it, the banks would be free of that implicit tax and the money would go to them instead of the Treasury. Contrariwise, banks (the agents) have a clear motivation to avoid maintaining unnecessarily large reserves.

A new strategy, the retail sweep program, was first initiated in 1994 by a North Carolina bank. Under this scheme, banks move forecasted excess funds out of checkable deposits into money market deposit accounts that have no reserve requirements. It could not have been clear at the time how the bank's regulator, the Federal Reserve, would respond. To date, the Federal Reserve has neither sanctioned nor banned such schemes. Anderson and Rasche [1] state that retail sweep programs have led to an extraordinary unwinding of statutory reserve requirements in the US, reducing required reserves in December 1999 by an estimated $34.1 billion.

Another illustration in the world of finance is the practice of short selling, the sale of stock that one does not yet own; this is profitable if the price of the stock later falls. In

comparison with "long" selling, where one makes a profit if the price of the stock later rises, selling short is particularly open to manipulative and deceptive techniques. This can give rise to cycles of loophole discovery and closure between brokers (the agents) and regulators (the legislature) – *cf.* Finnerty [8].

Ideally, the legislature has the goal of eliminating manipulative behavior, in this context the "intentional interference with the free forces of supply and demand" [8], leading to artificially depressed share prices. In lieu of this inaccessible $\mathbb{V}$, various abridgements are used. Historically, the British parliament banned short selling altogether between 1734 and 1860, while the U.S. Securities Exchange Commission's "uptick rule" (1934) permits short selling only following an uptick (trade where the price of a stock rises).

Unlike these financial scenarios, academic tenure, our final example, involves a *decision procedure*. Conceptually, agents are continually being removed from the space (a fixed interval after their creation) and replaced with new agents at "random" initial positions. Tenure is presumed to be granted if an agent's $\mathbb{A}$-value is above a fixed constant at termination.

The legislator wishes to make a decision based on the long-term research potential and teaching ability of the assistant professor. Typical proxies are teaching evaluations, letters of recommendation, and candidate-compiled research dossiers. It has been argued, somewhat dubiously, that "the minimum publication requirement of the tenure contract induces the optimal level of research with less variation in expected income, avoiding inefficient behavioral responses to the greater riskiness of a contract rewarding only realized publications" [6]. However, it seems unavoidable that abridgements based on proxies such as number of publications may cause legislative misdirection away from the hard-to-measure cost and risk of innovation.

7. Conclusion

In this paper we have developed a simple conceptual and mathematical framework for describing a legislature incentivizing a population of agents toward complex goals. We describe some qualitative phenomena of interest exhibited in by the implementation of a simple version of the framework. The relationship to real-world phenomena is also explored.

A taxonomy of agents is given, distinguishing between agents optimizing in socially productive and socially unproductive ways (as seen from the legislator's point of view). The results of simulation are given, in which the rapidity of legislative revision and the rapidity of agents' strategy revision are in competition. The model is shown to produce societies in which agents behave pathologically, as well as societies in which agents can be effectively controlled, exactly as one would hope from a simulation model with the right structural relationships.

References

[1] Richard G. Anderson and Robert H. Rasche. Retail sweep programs and bank reserves, 1994-1999. Technical Report 2000-023A, Federal Reserve Bank of St. Louis, St. Louis, MO, 2000.

[2] R.N. Bernard. Using adaptive agent-based simulation models to assist planners in policy development: the case of rent control. Technical Report Working Paper 99-07-052,, Santa Fe Institute, Santa Fe, NM, 1999.

[3] G. Boella and L. van der Torre. Local policies for the control of virtual communities. In *Procs. of IEEE/WIC WI-03*, Halifax, Canada, 2003. IEEE Press.

[4] G. Boella and L. van der Torre. Norm governed multiagent systems: The delegation of control to autonomous agents. In *Procs. of IEEE/WIC IAT-03*, Halifax, Canada, 2003. IEEE Press.

[5] C. Castelfranchi. The theory of social functions: challenges for computational social science and multi-agent learning. *Cognitive Systems Research*, 2, 2001.

[6] Zhiqi Chen and Stephen Ferris. A theory of tenure for the teaching university. *Australian Economic Papers*, 38:9–25, 1999.

[7] Joshua M. Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. The Brookings Institution, Washington, DC, USA, 1996.

[8] John D. Finnerty. Short selling, death spiral convertibles, and the profi tability of stock manipulation. *SEC Regulatory Action*, 2005.

[9] G.N. Gilbert and J. Doran. *Simulating societies: the computer simulation of social phenomena*. UCL Press, London, UK, 1994.

[10] H.L.A. Hart. *The Concept of Law*. Clarendon Law Series. Oxford University Press (Clarendon), London, 1961.

[11] Scott Moss, Thomas E. Downing, and Juliette Rouchier. Demonstrating the role of stakeholder participation: An agent based social simulation model of water demand policy and response. Technical Report CPM Report 01-79,, Centre for Policy Modeling, Manchester Metropolitan University, 2000.

[12] Jocelyn Pixley. *Emotions in Finance: Distrust and Uncertainty in Global Markets*. Cambridge University Press, Cambridge, UK, 2004.

[13] H.A. Simon. *Models of Bounded Rationality*. MIT Press, Cambridge, MA, 1982.

[14] Thorsten Veblen. *The Theory of the Leisure Class*. Number 1899 in History of Economic Thought Books. McMaster University Archive for the History of Economic Thought, 1899. available at http://ideas.repec.org/b/hay/hetboo/1899.html.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Invited Address:

Automated Analysis of Reasoning and Argumentation Structures in Texts

Helmut Horacek [1]
Fachrichtung Informatik
Universität des Saalandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
{horacek@cs.uni-sb.de}

Abstract. In many application areas of intelligent systems, natural language communication is considered a major source for substantial progress, even for systems whose pure reasoning capabilities are exceptional. Unfortunately, it turns out to be extremely difficult to build adequate natural language processing facilities for the interaction with such systems.

In this talk, I will expose some fundamental reasons for the difficulties associated with automatically analysing such inference-rich discourse, by elaborating discrepancies between effective human argumentation and efficient machine-based argumentative reasoning. On the human side, these discrepancies manifest themselves in several degrees of explicitness and levels of granularity, concise but ambiguous rhetorical signals, and varying conceptual perspectives, which all need to be related to uniform and fully explicit representations on the side of a machine. I will discuss approaches that aim at bridging these discrepancies to some degree, for both analysis and generation of argumentative texts. Issues addressed include disambiguation methods for discourse markers, identification of expected arguments, and dedicated content planning and structuring techniques.

[1]Correspondence to: Helmut Horacek, Fachrichtung Informatik, Universität des Saalandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany. Tel.: +49 681 302 2450; Fax: +49 681 302 5076; E-mail: horacek@ags.uni-sb.de.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Automatic Translation from Textual Representations of Laws to Formal Models through UML

Pietro Mercatali and Francesco Romano
ITTIG-CNR, Via Panciatichi, 56/16 - 50127 - Florence, Italy
{mercatali, romano}@ittig.cnr.it

Luciano Boschi and Emilio Spinicci
Dipartimento di Sistemi e Informatica – Università degli Studi di Firenze, Italy
{boschi, spinicci}@dsi.unifi.it

Abstract. A combination of UML and text mining, or more in general Information Extraction, can provide a valuable help to people involved in research about the linguistic structure of statutes, and, as a side effect, can be the seed for a new generation of applications in the legal domain. In particular, in this paper we present LexLooter, a prototype for amendments modelling and legislative text co-ordination based on UML and Natural Language Processing.

Keywords. Amendments, Legimatics, LexEdit, LexLooter, NLP, UML

1. Scope

We present in this work tools and techniques for the generation of a formal representation of legislative documents using UML modelling [1,2,3] with the purpose to represent connections between normative dispositions inside different normative acts. Such techniques exploit NLP for information extraction and syntactical and semantic tagging via XML markup of the elements of a normative text. The structural tagging allows to build a UML model representing the external structure of legislative acts (articles, paragraphs, commas, letters, etc.), while the text embodied inside each tag is used to instantiate each modelled element.

In this context, we have applied this modelling technique to Express Text Amendments (ETAs), referring to a given normative text, for the compilation of the coordinated text through the corresponding UML model.

The readability appointed to the *coordinated model* by UML improves the processing of modification applied to the original normative text by its ETAs, thus constituting an added value for the representation and the interpretation of the legislative text.

This paper is structured as follows: legal rules that constitute the basis for the UML representation of normative acts (and in particular of Express Text Amendments) are detailed in sections 2 and 3; section 4 briefly introduces NLP tools for legislative information markup, developed by the authors in previous works [4], [5] on which the prototype presented in this paper is based; section 5 details the UML model instantiated by LexLooter (described in

section 6), the tool implemented for the generation of the coordinated model and text; in the final section, perspectives and potential applications of the methodology will be discussed, to contribute to the drafting, the communication and the interpretation of the legislative acts.

2. Identification and Description of the Models on the Basis of Legal Rules

The legislative instrument has, by definition, a prescriptive function, or in other words, it influences the behaviour and status of the addressee, who cannot escape from this function.

In virtue of this, the request (it is based on the same principle of representative democracy which legitimates and, at the same time, binds those who draft legislative acts, namely, the legislator) is that the legislative discourse relies on a set of rules that dominate and, at the same time, stand beside, integrate, and sometimes modify the rules that make up natural language, in our case Italian.

These rules can be defined as legal rules. The category is broad and varied; it includes within it rules with prescriptive force, that vary remarkably on the basis of the source from which they come, to whom they are addressed and the sanctions they bring with them, etc..

Concerning the enforcing and effectiveness of the models constructed on the basis of these rules, we would like to add several considerations. We are able to list three characteristics on which their efficiency largely depends:

1. <u>Flexibility</u> of the model, which must be adaptable to the many structural, functional and thematic characteristics of legislative instruments, which use that extremely changeable and unpredictable vehicle known as language.
2. <u>Precision</u> in the definition of the model itself, thanks to the presence of the legal rules which are in many cases detailed and precise, and, for the prescriptive nature of the legislative instrument, it goes without saying that reconciling flexibility and precision constitutes a crucial point in the construction of models.
3. <u>Authoritativeness</u> with which the model must be endowed, in order to be shared and held to be valid by all the users of the system. It is clear that this authoritativeness is guaranteed more if the model is derived from the legislative rules, in the strict sense, according to their hierarchy, while it will be less guaranteed, for example, by those rules which only the drafter must follow. It could be more or less guaranteed, to a different extent, by models based on consolidated practices, legislative theories shared by the major part of legal authority, and so on. We can, nevertheless, say that while could greater precision correspond to “more authoritative” models, greater flexibility should correspond to "less authoritative" models.

It should further be noted that very authoritative models have greater communicative value (they are directed at and basically are accepted by everyone, because they are enforced by strong and precise rules), whilst as authoritativeness gradually decreases, the models assume an always increasingly interpretative value, that is, they are a “subjective” definition of the function of the single textual structures or of the entire legislative instrument.

If we can, therefore, talk about “Normativistic Models” and “Interpretative Models”, it becomes fundamental to evaluate how we want to use them, and consequentially, refer to whether to the former or the latter.

For example, for the description of a legislative instrument, for the purpose of communication *erga omnes*, we cannot fail but to refer to a “normativistic model” while the use of an “interpretative model” could be dangerous and misleading.

It should be also specified that the models supposed to be defined for the corresponding

UML formalization can refer either to the whole legal system or to individual acts or dispositions, as well as elements inside each disposition.

Such models are however constituted by textual elements and do not allow the formalization of extra-textual elements implying the interpretation of the text on the basis of real world situations. So, we can talk about a normativistic approach rather than an interpretative approach, directed therefore to investigate the unity, consistency and completeness of the modelled element (legal system, act, and so on) rather than assumptions and premises or implications and, particularly, consequences of such an element [6].

3. Express Text Amendment or Novella: Definition of the Structure and of the Constituent Elements

We propose the modelling of text amendment dispositions, both because this fits for the particular application of this study, and because the Express Text Amendment or *Novella* is a structure peculiarly bound with linguistic and the legislative technique rules used for its definition. Moreover, the treatment of amendment dispositions is also subject of analyses aimed to the reconstruction of the normative coordinated text [7].

Amendment provisions, according to Sartor, fall within the main types of legislative links "nessi", classified on the basis of their impact on the legal provision involved. Amendments distinguished from the other large branch of referrals or references, are legislative links characterised by the fact that the active provision affects the passive provision, eliminating it, changing the text or changing the legal significance (whilst leaving the text unchanged). This effect is, instead, lacking in the referral, where the active provision avails itself of the passive provision to complete its meaning, without influencing the latter [8].

In relation to the nature of the impact of the amendment on the passive provision, we distinguish between textual amendments, time-based amendments (that influence the period of time of the applicability of the passive provision), material amendments (that amend the legislative content of the passive provision without affecting the text). We shall only look at the first type, the express amendments of the text which, traditionally, lawyers in Italy name the *novelle*.

Indeed, it is perhaps more correct to say that the function of express legislative amendment is expressed through the following three aspects:

- the structure of the *novella*, made up of an introductory part, called *subparagraph*[1] and a part that contains the *express textual amendment;.*
- the characteristics of the amending legislative act and the amended act: indispensable for subsequently being able to reconstruct the amending links between the different legislative sources;
- the citation with which the document to be modified is cited, that expresses the legislative reference (also a textual reference), a fundamental element of the amending provisions.

On the basis of the three aspects mentioned here, we have endeavoured to define and describe the qualifying elements of the amendment provision as detailed in the De Lorenzo's assay [9] and in the authors' previous works [11].

[1] Understood as the 'part of the provision that introduces the amendment': it contains the purview aimed at specifying the relationship (substitution or integration or abrogation) between the provision in force previously and that provided by the textual amendment. The new sub-section generally ends with a colon, followed by the textual amendment placed between inverted commas.

Once the elements making up the amending provision had been identified and described [11], we have been able to propose a classification based on two of the elements we believed to be particularly important: the action of amending and its object.

In particular, on the basis of the action of amending, a distinction can be made among the following: repeal, integration and substitution.

As far as the object is concerned, the amendment, instead, operates on either a part (supra-part, article, paragraph, etc.) or on a part of the legislative discourse. It is obvious that each of the identified actions can operate on both the object *"part"*, and on the object *"part of the discourse"*.

The model currently implemented does not take into account elements (both textual and extra-textual ones) that may influence the amendment action unfolded by the novella upon the modified text. In particular, we have assumed as default the period of being-in-force equals to the one of the modifying act, even if this does not prevent the system to be afterwards able to treat a different being-in-force period of the amendment disposition, with respect to the general one of the modifying act.

Instead, according to the normativistic approach previously introduced in section 2, the implemented system is not able to treat different effectiveness periods, because such elements are strictly dependent upon extra-textual aspects which are not included in the model[2].

4. Tools for Legislative Information Markup

For the UML modelling of Express Text Amendments, we need the extraction of semantic information for the recognition and markup of the amendment disposition, as well as its composing elements [11].

To locate the *novella* in the modified normative text, the extraction of structural information, for the markup of the partitions constituting the structure of the normative act (articles, commas, letters, and so on), is also needed, in order to generate some kind of mapping of the text.

For this last function we have used the structural analyzer implemented in the prototype of LexEdit XXI[3], while the extraction of semantic information, needed for the recognition of the amendment disposition, is provided by Sophia 2.1 Linguistic Parser by CELI of Turin, Italy[4]. Both tools will be briefly introduced in the following subsections.

---

[2] The difference between force and effectiveness is often exemplified [10] by the war criminal code: in peace time the code is certainly enforced but not effective. The effectiveness will depend from wartime declaration, but also from real situations not calculable by a representative model of legislative system because extraneous to the system itself. The war criminal code considers, in fact, cases of "automatic" application of war criminal law connected to some events, without the needs of a bill, order or other and, in particular, the war declaration which enforced, for general rule, the application of the war criminal code (art. 3 c.p.m.g.).

[3] One of the first legimatic software programs was LexEdit which, specifically, developed checking functions regarding the correctness of the text.

A new project about evaluating and checking the legislative text, is now underway at ITTIG (Istituto di Teorie e Tecniche dell'Informazione Giuridica) for basing the checking functions, typical of LexEdit, on those of a parser making the recognition of the legislative language much more efficacious. The first release of the project is the prototype LexEdit XXI [5].

[4] See http://www.celi.it/sophia.htm

4.1. Sophia

The parsing system Sophia 2.1 has been identified as a suitable tool for the recognition and tagging of the amendment provisions[5].

Sophia uses the techniques of finite state automata and finite state transducer, which make this software flexible and configurable, thus enabling rules and specific models (already defined or in the course of definition) to be formalised. In particular, we are working with this software on analysing and tagging the first sample of legislative instruments in the following phases:

- normalisation of the entry text, properly tagging all those structures and textual segments that can be recognised on the basis of characters or, in other words, without resort to or consultation of the lexicon-dictionary;
- lexical (syntactical category) and morphological (flexion passages) analysis of the text in input;
- disambiguation of the syntactical category of the words (*Part of Speech Tagging*);
- partial syntactical analysis (called *chunking*), aimed at identifying the minimum syntactical groups present in the text in input and at grouping them in constituents;
- semantic analysis and identification of the relevant conceptual structures in the text in input;
- conversion of the analysed document from the original format (Microsoft® Word, HTML, RTF, txt, etc.) into the XML format, according to the established DTD.

The compilation of the grammar in the syntax of the chosen parser takes place by using the semantic module of the system, through drafting legislative rules, that formalise the defined models of "novella", and permit the automated identification of the described linguistic structures and the information extraction.

4.2. The Zoniser of LexEdit XXI

LexEdit XXI is a tool composed by several modules. Among them, we have used for this analysis the **Zoniser** [5], whose purposes are:

- The identification of the structure of the document in terms of articles, paragraphs, etc.
- The production of a XML document in output according as far as possible to the DTD of NIR[6].
- To point out to the user any anomalies or decisions that cannot be made automatically.

The Zoniser receives as input a document in RTF format, which is validated according to the NIR DTD. After the validation step, the document is saved in a XML file containing the structural tagging; such output will be further exploited to build the UML representation of the processed normative act.

---

[5] Considered the general reliability of the linguistic parser, we have used one of these tools whose syntax we had already implemented with specific rules of the legal domain.

The most recent tools integrating the techniques of symbolic analysis (using dictionaries, grammars, thesaura, etc.) with the statistics analysis algorithms seem to distinguish themselves for the greater speed and efficiency; for the experimental and methodological goals of the proposed research, we have assessed the reliability a decisive characteristic.

[6] See the *Norme In Rete* (Law On The Net) *NIR* project: http://www.normeinrete.it

5. UML Model for Normative Acts

Once defined the scope and the target of this work, we need to formalize the subject of the analysis through the use of UML language; in other words, we need to define a model expressing both the structural aspects of a legal text (articles, commas, etc.) and the semantic ones (that is, the amendment information a novella applies upon an existing legislative text).

Structural aspects have been defined taking into account the NIR DTD in its loose version [12,13] and the information of structural correction provided in LexEdit XML output. The determined elements and their associations are represented in the diagram of Figure 1. According to the adopted DTD, all the elements belong to the NIR package: the main Class is *Legge* (Law) which includes several attributes, recording the identifiers of the normative act, as well as emission, publication, and the enforced date. Such class can be connected to a unique *Articolato* (Articles) class, linked on its own to one or more *AbstractPartition* classes (a generic partition of the Articles). The Articles class has been modelled to conform to the NIR DTD. A partition can be one of the sections included in the Articles, such as *Libro* (Book), *Parte* (Part), and so on. All of them are identified by a *num* attribute (the index of the partition, inherited by *AbstractPartition*) and can be associated to Articles.

Finally, each kind of partition may include other partitions: this is needed to model the nesting of the legislative structure (e.g. a *Titolo* is composed by one or more *Articolo,* which may include several *Commas*, which may include letter elements *EL*).

Moreover, according to the loose version of the NIR DTD, the proposed model allows also for the validation of structures that can be considered as incorrect, referring to the most recent rules of legislative drafting; for example, instances of *Sezione* could be used as subpartitions of an article, or different kinds of partitions could be combined together. This makes the model more flexible and able to treat also normative text preceding the adoption of the NIR DTD.

In the same way, text components have been modelled using a generic *Elemento* class, (which can be, for example a *Rubrica*) containing in a text attribute the body of the element.

The model previously described represents the formal structure shared by the legislative texts. Express Text Amendments extends the structural model with the elements defined in De Lorenzo's classification of ETAs: according to this definition, two other kinds of partition have been added to model *integrations*, *substitutions* and *repeals*: *IntegrazionePartizione* and *IntegrazioneStringa* have been included to model the integration or substitution respectively of an entire partition, and of single words or sentences; similarly, *AbrogazionePartizione* and *AbrogazioneStringa* have been included to model the corresponding repeal actions. Inside these classes, suitable text attributes identify, where needed, the parts of text to be modified and the modifying ones. The *qualifier* attribute is used to locate the position inside the partition where the amendment will be applied to. The *Riferimento* (Reference) class is used to detect the partition which the amendment provision will be applied to. Finally, instances of the *Change* class are added in the model describing the text in force, to take into account all the amendments applied over a period of time. The needed information to instantiate the previous additional elements is extracted from Sophia XML markup; the obtained UML elements are added to the structural model of the ETA, by linking each amendment to the partition including it.

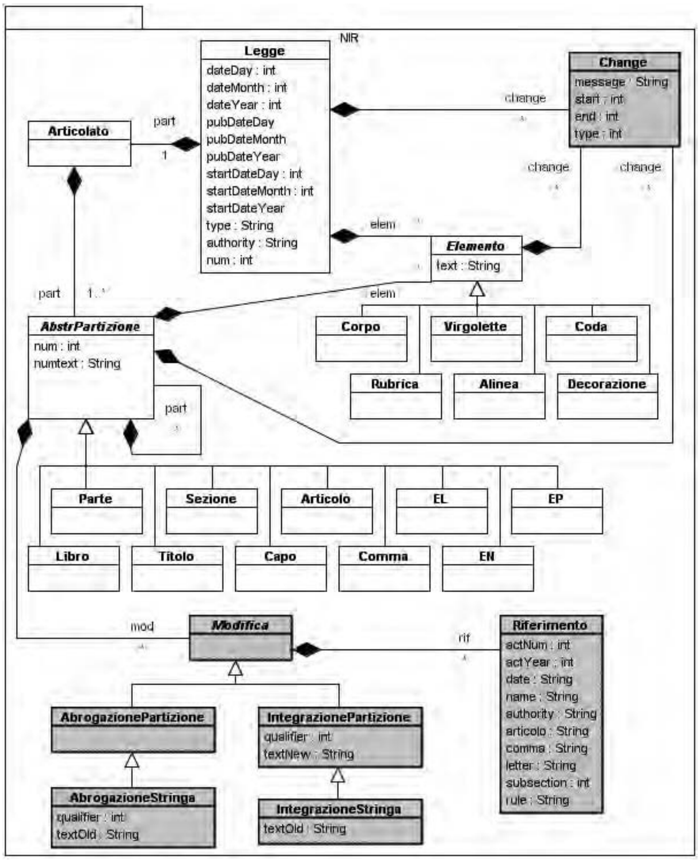

Figure 1. UML model of a legislative act with structural (in white) and amendments elements (in grey).

6. LexLooter Plug-in

LexLooter is a plug-in designed for the Eclipse 3.0 framework which is able to generate a UML representation of a legislative document, both in the original version and after the application of Express Text Amendments (ETAs), using the NIR DTD rules for the extraction of the articles structure from the legal text, and the model of the *novella* detailed in section 3.

Related to a given text of a normative instrument, LexLooter receives two kinds of XML input documents: the original legislative text with the XML markup of articles structure introduced by LexEdit XXI, and the XML files of Express Text Amendments, one including LexEdit articles structure of the novella, and another one with the XML tagging of amendment actions provided by Sophia. To generate the UML representation of a legislative document, LexLooter executes a sequential parsing of the marked up text: for each of the structural tags found, an instance of a suitable UML object is generated in the model (e.g. for each <ARTICOLO> tag, an instance of the UML class "Articolo" is generated); while the text embodied between corresponding tags is used as value for the attribute belonging to the object identified by the tag (e.g. the text corresponding to a <RUBRICA>).

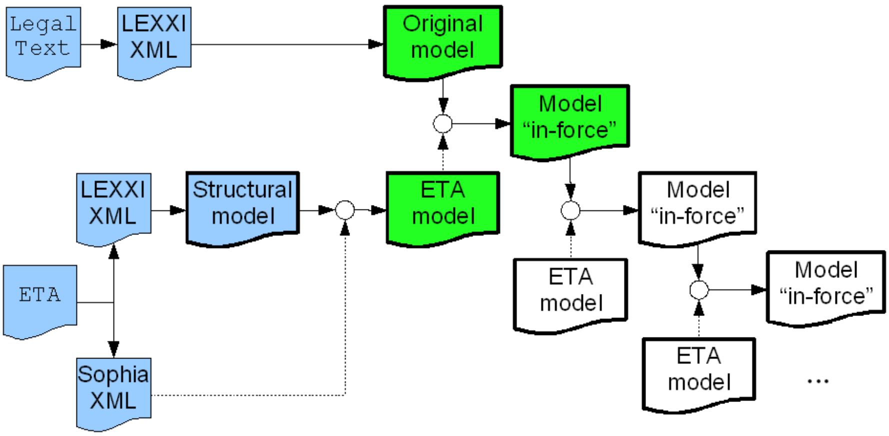

Figure 2. LexLooter processing chain of original law and ETA text.

A UML representation of Express Text Amendments can also be generated using LexLooter; in this case, the model is generated by exploiting also the text of *novella* processed by Sophia, whose markup allows to locate and extract all the amendment actions.

Once obtained the UML model of a *novella*, the model and the corresponding text of the original normative instrument can be updated by applying all the amendment actions previously extracted; the current law-in-force model and text can be finally obtained if applying iteratively each ETA to the original text of the law (Figure 2).

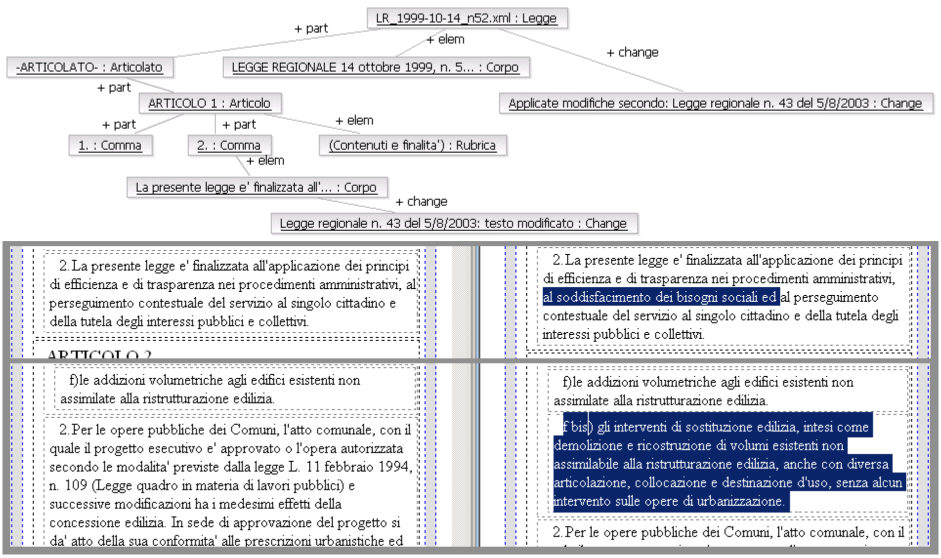

Figure 3. Coordinated UML diagram and text produced by LexLooter.

Figure 3 shows a partial UML diagram of the coordinated model and text produced by LexLooter: the UML object diagram represented in the upper part of the figure has been generated by instantiating the elements composing the structure of the original text; amended elements are associated to one or more *change* elements which point out the kind of applied modification.

More in detail, the lower part highlights the coordinated text after the application of an *insertion* amendment action (right) to Comma 2, shown in the diagram, compared with the original text (left). The availability of a UML object diagram of the amended text, combined with the generation of the coordinated text, allows for a very easy text consultation and a valid tool to verify or simulate the effects of the amendments.

LexLooter allows also for the generation of a Gantt diagram to trace the periods of enforcing of the normative instrument after the application of each ETA.

Table 1. Results of ETAs modelled by LexLooter applied to L.R. 52/1999.

L.R. 52/1999 (2 Novellas applied)	*Express Textual Amendments*					
	Marked by Sophia			*Applied by LexLooter*		
	Str. Repeal	*Str. Subst.*	*Str. Ins.*	*Str. Repeal*	*Str. Subst.*	*Str. Ins.*
	Part. Repeal	*Part. Subst.*	*Part. Ins.*	*Part. Repeal*	*Part. Subst.*	*Part. Ins.*
Number of modelled ETAs	***42***			***39***		
	0	9	4	0	9	2
	9	15	5	9	15	4

Table 1 shows some figures related to a significant example of LexLooter application to a normative text (namely, Regional Tuscany Statute 52/1999), and to its 2 corresponding ETAs: the figures point out that the number of ETAs modelled in the overall by the tool (39) correspond to the 92% of the ETAs that has been marked by the parser Sophia (42).

Two cases have been currently experienced for missing modelling:

- Typing errors in the text of the *Novella*, which result in incorrect extractions of the textual amendments. Current implementation normalizes to ASCII characters the text to be extracted, in order to optimize the matching with the set of adopted regular expressions; these will also be further refined in next releases to improve amendments extractions.
- Missing LexEdit XXI structural markup of text inserted by an amendment, which appears commonly quoted in an ETA; this prevents LexEdit to recognize possible added partitions (i.e. one comma substituted by more commas); the missing structural markup of the modifying text results in the generation of an incomplete coordinated model. Next releases will include a better integration with the structural marker, to generate a full model of a coordinated text.

Moreover, we have also experienced a number of amendments that have resulted not marked by Sophia: though able to estimate such amendments (via suitable warning messages indicating the number of occurrences in the text of modification actions outside Sophia markups), LexLooter cannot model them. This is due to the fact that Sophia can mark only *well-formed* amendments, accordingly to the modern drafting rules that have been translated in the extraction model of the parser; by loosing such rules, Sophia will be able to mark also *bad-formed* amendments, thus allowing LexLooter to include them in a coordinated model. We also provide for the implementation in LexLooter of suitable verification algorithms in the coordinated model for the compliance of the amendments to the legislative drafting, either by the validation of the structure, if consistent with structural constraints (expressed with OCL) or with the insertion in the Change class of correctness attributes highlighting possible *bad-formed* amendments.

7. Work Perspectives

Though still at a prototypal stage, LexLooter is a new step based on previous authors' work about formalizing normative text, because it has shown the capabilities of the UML modelling to represent in an immediate way the relations between the elements inside a normative act. Provided the descriptive capabilities of the model are suitably extended, the tool can be employed for the reconstruction of the normative coordinated text by exploiting the particularly comprehensible and flexible UML graphical formalism.

Current work is in the direction of applying the methodology implemented in LexLooter to a larger corpus of normative acts, in order to verify with an extensive experimentation the efficiency of the system and the use cases.

This methodology for the representation of the normative text also opens up other interesting applications, in particular for the verification of the correctness, consistency and accuracy of the normative speech and of its composing elements. Even if we have referred to a model that can be defined as normativistic, we can however suppose a possible use of the system to support the interpretation, in order to highlight what can be defined as "critical points" of the text (ambiguity, contradictions, etc.), where the interpreter has to make the most difficult choices. The UML notation also fits for the formalization of those extra-textual elements, which, in combination with the textual ones, provide a substantially complete set of information which the interpretation of the legal rule is based on; therefore, further research perspectives arise, relating to the representation of the real world into formalizable models; though fascinating, they currently take no part into this study.

Therefore, such methodology can be adopted for the implementation of suitable tools for supporting the generation, evaluation and consultation of normative texts.

Further extensions of this work could be in the direction of the application of the methodology to administrative documents, as well as any other acts where a formal model of the structure can be defined.

References

[1] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*. Addison Wesley, Sept. 1998.
[2] J. Rumbaugh, I. Jacobson and G. Booch, *The Unified Modeling Language Reference Manual, 2nd Ed.* Addison Wesley, Jul. 2004.
[3] T. M. van Engers and E. Glassèe, *Facilitating the legislation process using a shared conceptual model*. IEEE Intelligent Systems, vol. 16(1), Jan. 2001, pp. 50-58.
[4] A. Bolioli, L. Dini, P. Mercatali and F. Romano, *For the Automated Mark-up of Italian Legislative Texts in XML*. In "Legal Knowledge and Information Systems. JURIX 2002: The 15th Annual Conference". IOS Press, 2002.
[5] P. Mercatali, F. Romano, *Automatic legislation evaluation*. In Proc. of ICAIL'05, Bologna, Italy, Jun. 6, 2005.
[6] G. Carcaterra, *Metodologia giuridica*. In *Corso di studi superiori legislativi*, Cedam, Padova, Italy, 1990.
[7] M. Palmirani, R. Brighi, *Norma-System: A Legal Document System for Managing Consolidated Acts Database and Expert Systems Applications*. In Proc. of DEXA 2002, Aix-en-Provence, France, Sept. 2-6, 2002.
[8] G. Sartor, *Riferimenti normativi e dinamica dei nessi normativi*. In *Il procedimento normativo regionale*, Cedam, Padova, Italy, 1996.
[9] M.C. De Lorenzo, *Modelli di novelle*. In *Informatica e diritto*, vol.1, 2002.
[10] G.U. Rescigno, *L'atto normativo*. Giappichelli, Torino, Italy,1998.
[11] A. Bolioli, L. Dini, P. Mercatali and F. Romano, *Models of "Novelle" and Normative Grammars*. In "Legal Knowledge and Information Systems. JURIX 2004: The 17th Annual Conference", IOS Press, Amsterdam, 2004.
[12] C. Biagioli, E. Francesconi, P.L. Spinosa and M. Taddei, *The NIR Project. Standards and tools for legislative drafting and legal document Web publication*. In Proc. of the Int. Conf. of Artificial Intelligence and Law, Edinburgh, Jun. 24, 2003.
[13] A. Marchetti, F. Megale, E. Seta and F. Vitali, *Marcatura XML degli atti normativi italiani. I DTD di Norme in Rete*. In *Informatica e diritto*, vol.1, 2001, pp. 123-148.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Some Foundational Linguistic Elements for QA Systems: an Application to E-government Services

Farida Aouladomar [1]
IRIT, France.

Abstract. Time saving and time flexibility of eGovernment procedures is more attractive than face-to-face services to citizens. Citizens may interact with government via emails, search administrative information via eGovernment portals, or even via large-public search engines. Procedural question-answering systems are of much interest to query legislation, court decisions, guidelines, procedures, etc. In this paper, we present a typology of how-questions asked on the web. Then, we explore facets of procedural texts : their typology and general prototypical structures. We finally present our strategy for answering procedural questions using the notion of questionability of a procedural text.

Keywords. Procedural Questions, Procedural Texts, Question-answering Systems, eGovernment Services

1. Introduction

Procedural questions, sometimes called ’How-questions’, are questions whose induced response is typically a fragment, more or less large, of a procedure, i.e., a set of coherent instructions designed to reach a goal. Procedural questions form a large subset of questions typically introduced by ’How’.

Recent statistics elaborated from queries to Web search engines show that procedural questions is the second largest set of queries after factoid questions [10]. Procedural question-answering systems are of much interest both to the large public via the Web, and to more technical staff, for example to query large textual databases dedicated to various types of procedures.

Such systems are of particular interest eGovernment services since they are increasingly growing. For example, the online income tax application form was so successful in France that the government extended the deadline for internet users only. Governments are today aware of the significant benefits that can be realized by migrating traditionally face-to face services to the Internet. Administrative procedures are thus accessible via portals which are now emerging as the new e-Government single points of access for citizens and businesses [1]

[1]Correspondence to: Farida Aouladomar, IRIT, 31062 Toulouse Cedex, France. Tel.: +33 61 55 74 34; E-mail: aouladom@irit.fr.

The work presented here is a generic study designed to answer general procedural questions on the web. The answers to "how" questions can be found in procedural texts. Under the heading of procedural texts, there is a quite large diversity of texts (receipes, maintenance manuals, advice texts, etc.) [2] notes the variability of judgements in procedural text categorization, depending on the text main objectives and style. Procedural texts represent a substantial part amoung legal texts. They take the form of guidelines, circular letters, legislative texts and instructions of governmental bodies, etc.

Procedural texts share general common structures but each type of text has its own characteristics. Administrative texts are, for example, richer in conditions and contain almost no picture. Whereas assembly notices are richer in arguments, pictures, etc.

Procedural texts explain how to realize a certain goal by means of actions which are at least partially temporally organized. A procedural text can indeed be a simple, ordered list of instructions to reach a goal, but it can also be less linear, outlining different ways to realize something, with arguments, advices, conditions, hypothesis, preferences. It also often contains a number of recommendations, warnings, and comments of various sorts.

Procedural texts adhere more or less to a number of structural criteria, which may depend on the author's writing abilities and on traditions associated with a given domain. The administrative and legal domains have for instance their own writing techniques and layout. Typographical conventions such as the use of hyphens, bullets or other forms of numbering in front of each of the enumerated instructions are often imposed to writers via e.g. style files. The same is observed for a number of editing recommendations: language level, size of sentences, pronominal references, etc.

From a methodological point of view, our approach is mainly based on a corpus-based analysis, whose aim is to extract the general structure of procedural texts that is tailored for question-answering. Our corpus contains various types of procedural texts including texts as different as receipes, administrative procedures, maintenance/assembly notices, advices texts, rules, etc. In our corpus, we focused on procedural administrative texts for eGovernment documents. To get rights, citizens must often fill in forms, specifying various information. The requested information reflects the current status of the legislation and may be quite complex [7]. Application forms include, therefore, notices that help the applicant to fill them out.

In this paper, we first propose a general typology of procedural questions. In section 2, we develop an analysis of procedural texts, from the point of view of their discursive structure. In section 3, we show and evaluate the adequacy of this analysis for answering How-questions. For that purpose, we introduce different notions, among which the notion of the $questionability$ of a text, i.e. its ability to be used to answer How-questions. Finally, in section 4, we sketch out a few procedures to retrieve relevant sets of instructions from How-questions .

2. A Typology of How-Questions

Let us now investigate the structure of procedural questions. Besides an introspective analysis, the work reported below is largely based on corpora studies. We considered in particular FAQ which abound in procedural questions, questions in the TREC and AnswerBus frameworks, and quite comprehensive inventories built from queries submitted to search engines over the past month(s) composed of keywords, found at:

http://inventory.overture.com/d/searchinventory/suggestion/?mkt=fr. We constructed our procedural question corpora from different domains : legal domain, health, education, tourism, social behavior (savoir-faire and others), computer sciences, maintenance.

In this section we first identify the nature of procedural questions, since they cannot just be identified by the interrogative pronoun *how* that introduces them.

A large number of procedural questions are introduced by the interrogative pronoun *Comment* in French (How in English, Wie in German, Cómo in Spanish, etc.). However, $Comment$, similarly to How, has several uses which are not all related to procedural questions : (1) **the boolean How**: *how are you ?*, (2) **the nominal How**: *comment dit-on maison en espagnol ? (how do you say 'house' in Spanish?)*, (3) **the causality How**: *how did John die?*, (4) **the instrumental or manner How**: *how is couscous eaten in Morocco?, response: by hand* (5) **the choice list How**: *how can I pay my air ticket?, response credit card, cash, etc...*, (6) **the instructional How**: *how to change my car wheel?*

Only the last type of How questions in this analysis is typical of procedural How questions. The instrumental, manner and choice list 'How-questions' may also contain some forms of procedures.

Procedural questions are often introduced by How, but there are several other forms that we survey below:

- Forms in **'Que + Faire'** (gloss: what to do to...), as in *what should I do to get a visa for India ?* or forms with 'Quel + ETRE + proposition' (which/what + BE + prop) as in *what are the steps to follow to get a visa for India?* . We sometimes find directly constructions using the noun *procedure* to express procedural queries as in *What is the procedure to obtain the french nationality?*.
- **Elliptical use of How**: abound in cases where just two or three keywords are used instead of a full, well-formed, natural language sentence: *find lawyer, lodge complaint*. The verb or the deverbal used in the query allows for the identification of the type of question. More complex elliptical forms encountered in our corpora include e.g. the need of inference to identify a goal from a problematic situation as in: *passport lost* which must be reformulated as *how to replace my passport?*
- Questions in **is it possible to, can I, etc. + VP**, as in *is it possible to get the dual nationality?*, have a direct response which is a priori just yes or no, but, in the case of a cooperative response, which is our perspective, the response is often a set of instructions, answering the question viewed as a goal to reach.

In our system, query processing is based on the QRISTAL system (developed by Synapse Toulouse : http://www.qristal.fr), which is not perfect, but it allows us to get the distinctions presented above and to construct an adequate representation.

3. An Analysis of the Structure of Procedural Texts

In this section, we introduce our analysis of the discursive structure of procedural texts, with the view (in terms of topics and granularity) of answering how-questions.

3.1. A Discursive Analysis of Procedural Texts

Here is, represented by means of a grammar, the structure we have elaborated for procedural texts. The structures reported below correspond essentially to the organization of the informational contents. Elements concerning the layout (e.g. textual organizers such as: titles, enumerations, etc.), and linguistic marks of various sorts are used as triggers or delimiters to implement this grammar.

In what follows, parentheses express optionality, braces are used when an element is compulsory but not always expressed in words, + iteration, / is an or, the comma is just a separator with no temporal connotation a priori, and the operator $<$ indicates a preferred precedence (i.e. the elements usually appear following the elements order given in the grammar nodes). Each symbol corresponds to an XML-tag, allowing us to annotate procedural texts in order to extract relevant informations for the response formulation.

The top node is termed **Text**:

Text → title, (summary), (warning), (pre-requisites), (picture)+ $<$ objective.

summary → title+.

Summary describes the global organisation of the procedure, it may be useful when procedures are complex (summary can be a set of hyper-links, often pointing to titles).

warning → text , (picture)+, (pre-requisites).

Warnings represent global precautions or preventions associated with actions or objectives (e.g. make sure you get all the required documents prior to any naturalization application): they may have a complex rhetorical and modal structure, and should be studied in more depth. At the moment, we consider a warning just as a text, which sounds sufficient in most question-answering (QA) situations. Warnings are in fact of interest for answering Why? questions.

pre-requisites → list of objects, instruction sequences.

Pre-requisites describe all kinds of equipments needed to realize the action (e.g. the different documents needed to fill in forms) and preparatory actions.

picture describes a sequence of charts and/or schemas of various sorts. They often interact with instructions by e.g. making them more clear. Analyzing this phenomena is outside the scope of this paper.

The objective level is the basis structure of procedural texts. It is often complex since it may contain embedded objective. Each group of instruction sequences is designed around a specific goal.

objective → {goal} $<$ instruction sequences+ / objective.

Instruction sequences is structured as follows:

instruction sequences → instseq $<$ {connector}$<$ instruction sequences / instseq.

Instseq is then of one of four main types below:

instseq →, imperative linear sequence / optional sequence / alternative se-

quence / imperative co-temporal sequence.

Each type of instruction sequence is defined as follows:

imperative linear sequence → instruction < {temporal mark}, instseq/ instruction. (e.g. *En ce qui concerne votre adresse à l'étranger, indiquez le nom de la rue et le numéro s'il en existe, et ajouter votre numéro de boîte postale si le courrier n'est pas distribué à domicile dans la localité où vous résidez. Précisez enfin le nom du pays.*) (gloss : *Concerning your address abroad, mention the street name and the number if it exists and add your postal box number if mails aren't delivered at the place where you live. Finally, specify the name of the country.*) An imperative linear sequence is the kind of most common instruction sequence in procedural texts. It can be composed of one or several instructions.

optional sequence → optionality expression, imperative linear sequence. (e.g. *Si vous préférez joindre les copies des documents requis, présentez les originaux le jour de votre rendez vous au service des visas*) (gloss : *if you prefer enclosing copies of requested documents, bring originals the day you come to the visa services.*)

alternative sequence → (conditional expression), imperative linear sequence, (alternative-opposition mark) < instseq / (conditional expression, instseq)+. (e.g. *pour expédier votre demande, vous pouvez soit l'adresser directement par la poste, de préférence en recommandé, à la mairie de la commune où vous désirez vous inscrire, soit la confier au consulat, qui l'acheminera par la valise diplomatique, à l'adresse de la mairie.* (gloss : *to send your application, you can either post it, preferably in certified delivery, to the town council where you want to register, or give it to the consulate which will send it to the town council using the diplomatic bag*).

imperative co-temporal sequence → imperative linear sequence < co-temporal mark < imperative co-temporal sequence / imperative linear sequence.
A co-temporal sequence relates instructions which must be realized at the same time, or more generally non-sequentially (e.g. *Pour bénéficier des services de l'ANPE, vous devez vous inscrire à votre agence locale, et parallèlement faire une demande d'indemnisation aux Assedic*) (gloss : *to pretend to ANPE services, you should registrer to your local agency, and at the same time, make a help request to the Assedic*).

Finally, Instruction is the lowest level and has the following structure, with recursion on objective:

instruction → (iterative expression), action, (goal), (reference)+, (argument), (picture)+, (warning).

At this level, it is most important to note reference phenomena of various sorts, pointing to already described instructions, to instructions to be described later, or to external data via hyper-links (e.g. *see application form 256...*). Let us also note that we have observed almost no restatements (other ways to describe an instruction if it is difficult to understand) and no summaries synthesizing instructions (not to be confused with the generic summary of an objective, which is a kind of table of contents).

3.2. Marks for Instruction Localization

The different types of marks that allow for the identification of the different elements presented in the grammar above are presented in [4]. In this section, we present the main marks used to define the boundaries of each instruction or set of instructions.

Typographic criterion: the next point is to isolate basic instructions, called 'instructions' in the above grammar, and within these instructions the 'action' itself. The problem is essentially to identify simple marks which are delimiters of the beginning and the end of an instruction. Inter-instruction marks organize instructions. They are in general quite simple to identify in procedural texts [11]. The beginning of an instruction is often the start of a sentence which can be introduced by various typographic marks proper to enumerations (intended lines, bullets etc.) [8]. These correspond also in general to an instruction. The end of an instruction is either a punctuation mark, usually the dot, sometimes the semicolon or the comma, or typographic marks introducing the next instruction. Table 1 summarizes our observations on the use of layout to easily identify the instructions within a procedural text. Communication category includes administrative texts, advice texts, etc.

Table 1. (1) percentage of instructions or set of instructions introduced by typographic marks such as hyphens, bullets and other numbering forms, line breaks. (2) number of instructions considered in our sample.

Domains	(1)	(2)
maintenance, assembly	78%	279
receipes	89%	151
communication	63%	206
average	77%	636

Semantic criterion: within an instruction, the action is in general organized around the action verb and its arguments. Goals, references, manners, limits, are all adjuncts which appear in various orders. Goals contain specific verbs while manners are often nominal. Using the same corpora as above, Table 2 shows the following verb distribution. It is elaborated with the TROPES software.

Table 2. (1) factive verb, (2) stative verb, (3) declarative verb, (4) performative verb.

Domains	(1)	(2)	(3)	(4)
maintenance/assembly	65%	23%	12%	
receipes	85%	13%	2%	
procedural QA pairs	67%	11%	22%	
communication	52%	26%	22%	
average	67%	18%	15%	
non-procedural texts	41%	35%	23%	1%

As can be noted, procedural texts have a much higher rate of factive verbs, and much less stative verbs. declarative verbs are about the same as in other types of texts. This

verb type discrimination criterion is not precise enough for procedural text categorization, in particular to discriminate communication procedural texts from non-procedural texts.

Morphological criterion: another criterion is the morphology of the verbs encountered in procedural text. Instruction verbs are usually at the imperative, infinitive and or gerundive form. The more these forms are found in texts the more procedural these texts are. Table 3 presents the results obtained over a large list of verbs for procedural text and non-procedural texts.

Table 3. (1) number of verbs in texts, (2) number of verbs in imperative/infi nitive/gerundive form, (3) percentage w.r.t. the total number of verbs.

Texts	(1)	(2)	(3)
maintenance/assembly	826	523	63%
receipes	434	386	89%
procedural QA pairs	495	244	49%
communication	315	156	49%
total average for procedural texts	2070	1309	63%
non-procedural texts	776	200	26%

Our goal is not just to use an instruction to respond to a How-question. It is often necessary to consider the level of the 'objective' or higher, where quite generic goals, those frequently encountered in How-questions, are found. 'objectif' are in general delimited by the expression of goals, which may have various forms, and by typographic elements (e.g. starting a new paragraph). Goals may be titles, well-identified from a typographic point of view, or they may have the form of a proposition introduced by a causal mark: *to vote by proxy, ...*

4. Questionability of a Text

We use the term *questionability*, term due to J. Virbel [9], to express the ability or the relevance of any text, in our case found on the Web, to respond to How-questions. The primary goal is to have criteria to identify texts which are procedural among those obtained from a search engine. The second goal is to consider those procedural texts which are the most appropriate for responding to procedural questions. This means to be able to compare texts clearly identified as procedural texts, in terms of their level of detail, informativity, readability, conciseness, illustrations, number of links to other pages or to other parts in that same text, prevention on actions, etc.

The evaluation of the questionability of a text can be made a priori, independently from any particular query, or in relation with a query since some goals may be easier to identify than others in given texts. In this section, we establish a compromise between these two views which are of much interest. For the moment, a response to a how-question is found in a single text, using relevance, clarity, and informativity criteria. In a second stage, it would be of much interest to select a text depending on the user profile (casual user or professional) and to be able to merge or to integrate texts when they complement each other. These objectives are obviously very difficult to implement.

Let us now present the different criteria we consider to measure the questionability of a text. This measure is decomposed into two stages. The first stage aims at selecting those texts which should be a priori procedural texts. It is essentially based on surface marks to guarantee a certain efficiency. The second stage concentrates on the query, and introduces several relevance measures correlated with the query to answer. It is presented in section 6.

The first stage, called the "CATEG", selects the subset of texts returned by a search engine which should be procedural texts according to the three 'surface' criteria below; measures are all relative to text size:

- typographic forms (noted as TF) of various kinds that measure the architectural quality of the text. These forms include those given in Table 1 of section 4.2.
- morpho-syntactic marks, (noted as MSM): in procedural texts, we observed (cf. Table 3, in 4.2) that most verbs are either in the infinitive or in the imperative form, there are also marks that motivate the user to go further, such as *you must, you just have to,* followed by an action verb, or marks that indicate a task to realize: *the next stage, the next step, proceed as follows, care about, do not forget to, etc.* which abound in procedural texts,
- the presence of a large number of articulatory marks, (noted as AM): temporal, argumentative, causal marks to cite the most important ones (section 4.2).

Since it is quite difficult to assign relative weights to each of these three criteria, we consider they have an equivalent weight in the selection of procedural texts. Each counts for a third of the decision. Given a set of n texts, we evaluate for each text TF, MSM and AM. For example, TF_i is the ratio: number of typographic forms divided by the size of the text in number of words in the text i. Then, for each criterion, the average frequency is computed:
$TF_{average} = (\Sigma_{i=1,n} TF_i)/n,$
and similarly for the other two criteria. We can now define the CATEG for text i w.r.t. the set of texts considered:
$CATEG_i = TF_i/TF_{average} + MSM_i/MSM_{average} + AM_i/AM_{average}.$

The second stage, the "QUEST", investigates in more depth the questionability of the text. The objective is to evaluate the number of areas which can potentially match with How-questions. This is carried out by identifying those areas in the text on which the matching with questions should potentially be realized. Via our corpora analysis, it turns out that those areas are essentially:

- the number of titles identified, under objectives and in the summary (noted as TIT),
- the presence of a large number of action verbs (noted as AV), (cf. Table 2, in 4.2),
- the number of goals identified, (1) associated with instruction sequences or (2) within basic sequences, associated with the action to realize, (noted as GOA), and, finally
- manners found in instructions (noted as MAN).

Different linguistic marks allow for the identification of the goals and manner : the causal and manner connectors(e.g. in order to, so, by + gerundive verb, with, etc.).

Similarly as above, we can define the QUEST rate for a given text i in a collection of n texts. $QUEST_i = TIT_i/TIT_{average} + AV_i/AV_{average} + GOA_i/GOA_{average} + MAN_i/MAN_{average}.$

We can then compute an estimate of the overall questionability of a text i in a collection of n texts as follows: $CATEG_{average} = (\Sigma_{i=1,n} CATEG_i)/n$,
$QUEST_{average} = (\Sigma_{i=1,n} NN_i)/n$,
$questionability_i =$
$CATEG_i/CATEG_{average} + QUEST_i/QUEST_{average}$.

5. Responding to How-questions

The main aim of this project is to adequately and cooperatively respond to How-questions. An accurate and relevant analysis of the structure of How-questions, of procedural texts and of the notion of questionability establishes a basis for associating a response with a query which is as adequate as possible. Within our present perspective, responding to how-questions involves the following tasks:

- selecting the procedural texts which have the best questionability rate. Since, at this level, the matching with the query has not yet been done, we keep the 20 best texts based on the metrics given above,
- matching the question body with 'questionable zones' of procedural texts, hierarchically organized as: titles, goals, manners, and defining the best match,
- extracting the relevant portion of the text and returning it to the user in a user-friendly way.

The last step consists in selecting the appropriate text fragment that responds the question. So far, our strategy is quite simple, and we have the following main situations:

- If the question matches with the title of the whole document, then the document is selected as a whole,
- If the question matches with the title or the goal of an instruction sequence, as defined in the grammar, then that whole sequence is selected. This is however a general rule which suffers some exceptions. In particular, for alternative sequences, it may be useful to select a larger fragment of the text.
- If the question matches with a goal within an instruction, then this instruction is returned to the user.

The adequacy of this rough strategy remains to be evaluated in depth. Since it is not easy to predict when a larger text fragment will be necessary, our strategy is to return a window that displays a priori the selected portion, however, the user can scroll it up or down to get a larger or a nearby text portion. Besides the response, in case of an indirect match (e.g. using more generic terms or synonyms), an explanation must be provided so that the user understand why he gets such as response. The explanation follows the template philosophy presented in WebCoop [6], outlining the terms that have been changed and why.

6. Perspectives

In this paper, we presented the general structure of procedural texts. We also investigated the structure of How-Questions, outlining those which really induce responses under the

form of sets of instructions. We then showed how a procedural text can be characterized using relatively external and simple criteria. Finally, we briefly presented how to characterize the questionability of a text, and how the response retrieval mechanisms can be constructed from this notion.

This work is still very experimental, it raises many questions. Our work needs to be deepened along many lines, including response accuracy (quality and scope), response generation, and the development of more elaborated evaluation methods, much more complex than e.g. the methods used in TREC for factoid questions.

Works on procedural question-answering systems has many applications in the legal domains and must be deepened and adapted to them. This work can support general public "how do I" questions for e-governments (administrative procedures) or also "how" questions dealing with court files, and court decisions for instance. Other legal-field documents such as legislation texts which have specific structure with specific linguistic marks and lay-out, must be investigated. Question answering systems in the legal field are both useful for large public and practicioners of the domain.

References

[1] Accenture report. eGovernment Leadership : Rhetoric vs Reality - Closing the Gap. http://www.accenture.com/xdoc/en/industries/government/2001FullReport.pdf, 2001.

[2] Adam, J.M.. *Types de Textes ou genres de Discours ? Comment Classer les Textes qui Disent De et Comment Faire*, Langages, 141, pp. 10-27, 2001.

[3] Allen, J., *Towards a General Theory of Action and Time*, Artifi cial Intelligence, 23:123-154, 1984.

[4] Aouladomar, F., Saint-Dizier, P., *An Exploration of the Diversity of Natural Argumentation in Instructional Texts*, 5th International Workshop on Computational Models of Natural Argument, IJCAI, Edinburgh, 2005.

[5] Aouladomar, F., *A Preliminary Analysis of the Discursive and Rhetorical Structure of Procedural Texts*, Symposium on the Exploration and Modelling of Meaning, Biarritz, 2005.

[6] Benamara, F., Saint-Dizier, P., *Advanced Relaxation for Cooperative Question Answering*, in: New Directions in Question Answering, in Mark T. Maybury, (ed), AAAI/MIT Press, 2004.

[7] The GIST Project, *Generating InStructional Text*, http://tcc.itc.it/history/projects/gist.html, 1994-1996.

[8] Luc, C., Mojahid, M., Virbel, J., Garcia-Debanc, C., Pery-Woodley, M-P., *A Linguistic Approach to Some Parameters of Layout: A study of enumerations*, In R. Power and D. Scott (Eds.), Using Layout for the Generation, Understanding or Retrieval of Documents, AAAI 1999 Fall Symposium, pp. 20-29, 1999.

[9] de Mattos Pimenta Parente, M-A., Steffen Holderbaum, C., Virbel J., Nespoulous, J-L., *Text Questionability as a predictor of story recall*, Thirteen Annual Meeting of the Society for Text Understanding, Madrid, 2003.

[10] De Rijke, M., *Question Answering: What's Next?*, the Sixth International Workshop on Computational Semantics, Tilburg, 2005.

[11] Takechi, M., Tokunaga, T., Matsumoto, Y., Tanaka, H., *Feature Selection in Categorizing Procedural Expressions*, The Sixth International Workshop on Information Retrieval with Asian Languages (IRAL2003), pp.49-56, 2003.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

A Question Answer System for Legal Information Retrieval

Paulo Quaresma [1] and Irene Pimenta Rodrigues [2]
Departamento de Informática, Universidade de Évora, Portugal

Abstract. In this paper we present a question-answering system for Portuguese juridical documents.

The system has two modules: preliminary analysis of documents (information extraction) and query processing (information retrieval). The proposed approach is based on computational linguistic theories: syntactical analysis (constraint grammars); followed by semantic analysis using the discourse representation theory; and, finally, a semantic/pragmatic interpretation using ontologies and logical inference.

Knowledge representation and ontologies are handled through the use of an extension to PROLOG, ISCO, which allows to integrate logic programming and external databases. In this way it is possible to solve scalability problems like the need to represent more than 10 millions of discourse entities.

The system was evaluated with the complete set of decisions from several Portuguese juridical institutions (Supreme Courts, High Court, Courts, and Attorney-General's Office) in a total of 180,000 documents. The obtained results were quite interesting and motivating and allowed the identification of some strong and weak characteristics of the system.

Keywords. Natural Language Processing of Legal Sources, Question Answering Retrieval in Law and Governmental Services

1. Introduction

This paper describes an ongoing project at the Informatics Department of the University of Évora, Portugal, aiming to develop a question answering system for the Portuguese language and to apply it to the juridical domain.

Question answering systems are an important topic of research in the natural language processing field and much work has been done by many researchers in the last years. Several international conferences have special tracks for this topic, namely, the TREC – Text REtrieval Conference (http://trec.nist.gov) or the CLEF – Cross Language Evaluation Forum (http://www.clef-campaign.org).

The legal domain is an area where question answering systems could (and should) be applied, allowing citizens to have an easier access to legal information. In the last years, some work has been done trying to develop such systems and, for instance, in the JURIX 2003 conference, a special workshop was made on the topic "Question Answering for Interrogating Legal Documents".

[1]E-mail: pq@di.uevora.pt
[2]E-mail: ipr@di.uevora.pt

In Portugal criminal processes are not kept in a way that facilitates the search for information. Most of them are kept just as a set of documents without any particular structure. In order to overcame the lack of a structured knowledge base, we propose the use of a question answering system with the following goals:

- Answering user questions posed in natural language using the information contained in the criminal processes.
  Using our system an investigator can interrogate the system obtaining useful information about:
  * Places: Where can we buy drugs?
  * Dates: When was Mr. X arrested?
  * Defi nitions: What is a drug dealer?
  * Specifi c: How many times was Mr X accused? Who was arrested by dealing drugs in process X? What crimes committed Mr Y?
- Indicate a set of relevant processes.
  The investigator may not interested in obtaining just answers to questions; it may be interested in the knowledge source used for answering those questions.
- Help in the task of semi-automatically structure the criminal processes.
  This task is important namely to enable the Portuguese State to obtain liable statistics about the problems of criminal investigation in Portugal.

The system has two modules: preliminary analysis of documents (information extraction) and query processing (information retrieval).

The analysis of the document collection and queries is done using models from computational linguistic theories. The methodology includes: syntactical analysis of sentences using the constraint grammar Palavras [3]; semantical analysis using discourse representation theory [5]; and, fi nally, semantic/pragmatic interpretation using ontologies and logical inference.

Knowledge representation and ontologies are handled through the use of an extension to PROLOG, ISCO[1,2], which allows to integrate logic programming and external databases. In this way it is possible to solve scalability problems like the need to represent more than 10 millions of discourse entities.

The QA system is able to answer queries in natural language, based on information conveyed by a collection of documents. The answer to a specifi c question is: a set of words and the identifi cation of the document and sentence, which was used as the source of information. For instance, for the following question:

Who is the President of Portugal?

Our system answers:

Jorge Sampaio - document: d125/doc040103/001 - sentence: 5

The proposed system is an evolution of a previous system evaluated at CLEF 2004 [6] and applied to a Portuguese newspaper domain.

In the next section the architecture of the system is described. In sections 3 and 4 the syntactical and the semantical modules are described in detail. Section 5 presents the knowledge representation approach. Section 6 describes the semantic-pragmatic interpretation of the documents, based on the previous analysis and on the ontology. Section

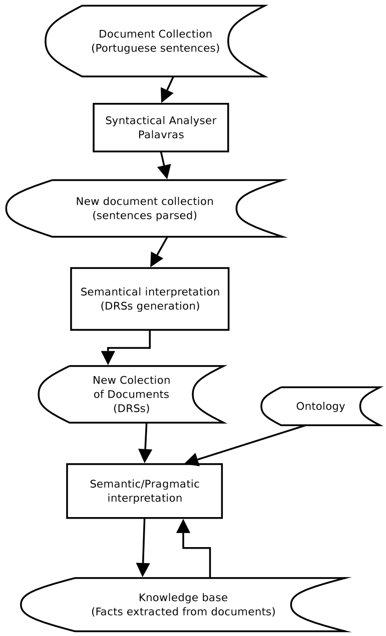

Figure 1. Document Processing.

7 shows the processing of a query and the generation of the correspondent answer. In section 8 the evaluation results are presented. Finally, in section 9 some conclusions and future work are discussed.

2. Architecture

The QA system has two main modules:

- Information extraction;
  This module extracts information from the documents and it creates a knowledge base. The module (see fi gure 1) is composed by several sub-modules:
  – Syntactical analysis: sentences are processed with the Palavras[3] parser. After this phase, a new collection of documents (with the parsing result) is obtained.
  – Semantic analysis: the new collection of sentences is rewritten [5] creating another collection, where each document has a DRS (structure for the discourse representation), a list of discourse referents and a set of conditions.
  – Semantic and pragmatic interpretation: in this phase the previous collection of documents is processed, taking into account an ontology and, as a result, a knowledge base is built. This knowledge base contains instances of the ontology.
- Information retrieval:
  This module processes the query and it generates the answer: a set of words and the identifi cation of the document and sentence where the answer was found. Figure 2 shows the diagram of this module. It is composed by the following phases:
  – Syntactical analysis: using the parser Palavras [3].

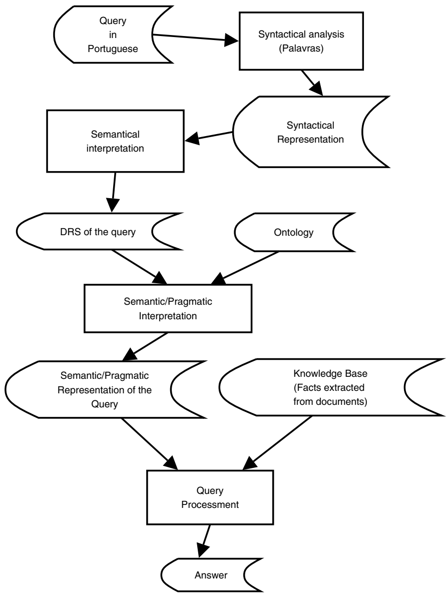

Figure 2. Query Processing.

– Semantic analysis: from the parser output, a discourse structure is built – a DRS[5] with the correspondent referents.
– Semantic/Pragmatic interpretation: in this phase, some conditions are rewritten, taking into account the ontology, and generating a new DRS.
– Query Processing: the fi nal query representation is interpreted in the knowledge base through the unifi cation of the discourse entities of the query with documents discourse entities (see section 7).

In the next sections these sub-modules are described in more detail.

3. Syntactical Analysis

Syntactical analysis is obtained through the use of the PALAVRAS parser from Eckhard Bick [3], developed in the context of the VISL[1] project at the *University of Southern Denmark*. This parser gives good morpho-syntactical information and it has a good coverage of the Portuguese language. For instance, in our system the verb lemma (infi nitive form) is used as the name of the predicates in the semantic analysis.

As an example, consider the following sentence (3.1):

A Joana morreu de asfi xia. "Joana died of asphyxia".

The syntactical structure of this sentence is the following:

```
sta(fcl,
  subj(np,
```

[1] Visual Interactive Syntax Learning

```
    n(art('o','F','S','<artd>'),'A'),
    h(prop('Joana','F','S'),'Joana')),
  p(v_fin('morrer','PS','3S','IND'),'morreu'),
  piv(pp, h(prp('de'),'de'),
    p(n('asfixia','F','S','<sick>'),'asfixia', '.')))
```

This structure is represented in Prolog and is used as the input of the semantic analyser.

4. Semantic Analysis

The semantic analysis rewrites the syntactical structure into a discourse representation structure [5], DRS. At present, we only deal with factual sentences, i.e., sentences with existential quantifi cation over the discourse entities. So, our discourse structures are sets of referents, existentially quantifi ed variables, and sets of conditions, fi rst order predicates.

Each syntactical tree, represented in Prolog, is rewritten accordingly with a set of rules and integrated in a DRS.

In order to allow alternative interpretations, the link between prepositional phrases in done using the relation *rel* with 3 arguments, the preposition and two discourse entities. This predicate *rel* allows the semantic/pragmatic interpretation to infer the adequate connection between the referents. For instance, the sentence 'O dono da casa'/'The owner of the house', is represented by the following DRS:

```
drs( entities:[ A:(def, male, sing),
                B:(def, female, sing)],
     conditions:[owner(A),
 house(B),
 rel(of,A,B)])
```

As it can be seen in the next section, this representation allows the semantic/pragmatic interpretation to rewrite the DRS, obtaining the following structure:

```
drs( entities:[ A:(def, male, sing),
                B:(def, female, sing)],
     conditions:[belongs(A,B), person(A), house(B)])
```

In order to show an example of a syntactical tree transformation into a DRS, we show sentence (3.1) rewritten :

```
drs (entities:[  A: (def, fem, sing),
                 B: (def, fem, sing) ],
     conditions:[ name(A, 'Joana'),
                  died(A),
                  rel (of, A, B),
                  asphyxia(B)])
```

User queries are also interpreted and rewritten into DRS. For instance, the question:

Como morreu Joana?/How did Joana die? (4.1)

is transformed into the following discourse structure:

```
drs(entities:[F:(def, fem, sing),
              G: interrog(que), male/fem, sing]
    conditions:[died(F),
                name(F,'Joana'),
                rel(of,F,G)])
```

This representation is obtained because “Como/How” is interpreted as “de que/of what”. In the semantic-pragmatic interpretation and in the query processing phase, the structure (4.1) might unify with sentence (3.1) and we may obtain the following answer: “Joana died of asphyxia”.

5. Ontology and Knowledge Representation

In order to represent the ontology and the extracted facts, we use an extension of logic programming, ISCO[1,2], which allows Prolog to access databases. This technology is fundamental to our system because we have a very large database of referents: more than 9 millions only for the Público newspaper.

Databases are defined in ISCO from ontologies. Our system uses two different ontologies:

- One ontology built by us aiming to model common knowledge, such as, geography (mainly places), and dates;
  This kind of knowledge is important to correctly extract facts from the documents and to be able to answer questions about places. The ontology defines places (cities, countries, . . .) and relations between places.
- One ontology generated automatically from the document collection [8,7];
  This ontology, although being very simple, allows the representation of the domain knowledge.

The ontology can be defined directly in ISCO or in OWL (Web Ontology Language) and transformed in ISCO [7].

The knowledge extraction module identifies facts (instances of an ontology classes) and inserts them as rows in database tables.

For instance, sentence (3.1), with semantic representation in page 5, would generate several tuples in the database. First order logical expressions are $skolemized$, i.e., each variable existentially quantified is replaced by a different identifier:

- `(123,''Joana'')` is added to table *name*
- `(123)` is added to table *die*
- `(124)` is added to table *asphyxia*
- `rel(of,123,124)` is added to table *rel*

In the document processing phase, our system uses the first sentence interpretation (note that for each sentence there might exist several distinct interpretations). This is caused by temporal and spacial complexity problems but it does not seem to decrease much the performance of the system. Nevertheless, additional measures should be done in order to fully evaluate the impact of this option.

Additionally, we also add information to the database linking referents with the documents and sentences were they appeared. For instance the tuple:
`(123,'d03/doc950605/005',4)` would be added to table *referred_in*.

6. Semantic/Pragmatic Interpretation

Semantic/pragmatic interpretation tries to reinterpret semantical information, taking into account the considered ontology.

This process receives as input a discourse representation structure, DRS, and it interprets it using rules obtained from the knowledge ontology and the information in the database.

In order to obtain a good interpretation, our strategy is to search for the best explanation that supports the sentence logical form. This strategy for pragmatic interpretation was initially proposed by [4].

The knowledge base for the pragmatic interpretation is built from the ontology description in ISCO. The inference in the knowledge base uses abduction and fi nite domain constraint solvers.

Suppose the following sentence:

"A. conduz com uma taxa de alcoolemia de 2.15."
"A. drives with an alcoholic rate of 2.15.".

which, by the semantic analysis, is transformed into the following structure: one DRS, four discourse referents, and a set of conditions:

```
drs(entities:[A:(def,male,sing),
              B:(indef,fem,sing),
              C:(indef,fem,sing),
              D:(def,male,sing)]
    conditions:[name(A, 'A.'),
                drive(A),
                rel(with,A,B),
                rate(B),
                rel(of,B,C),
                alcohol(C),
                rel(of,C,D),
                number(D,2.15)])
```

The semantic/pragmatic interpretation process, using information from the ontology, will rewrite the DRS into the following one:

```
drs(entities:[A:(def,male,sing),
              B:(def,male,sing)]
    conditions:[name(A, 'A.'),
                person(A),
                drive(A,_,_,_,B),
                alcohol\_rate(B,2.15)])
```

The interpretation of *rel(with,A,B)* as *drive(A,_,_,_,B)* is possible because the ontology has a class *drive*, which relates persons driving in a time interval with a alcoholic rate in blood.

One of the major problems of this phase is to correctly identify the distinct referents in the documents. It is important to use the same skolem constant to identify the same referent and different individuals should have always different identifiers (skolem constants).

7. Answer Generation

The generation of the answer is done in two steps:

1. Identification of the database referent that unifies with the referent of the interrogative pronoun in the question.
2. Retrieval of the referent properties and generation of the answer.

In order to illustrate this process, suppose the following question:

"Quem cometeu um homicídio por conduzir alcoolizado?"
"Who committed an homicide because he/she was driving drunk?"

This question is represented by the following DRS, after syntactical and semantical analysis:

```
drs(entities:[A:(who,male/fem,sing),
              B:(indef,male,sing),
              C:(indef,male,sing)],
    conditions:[committed(A,B),
               homicide(B),
               rel(because,A,C),
               drunk(C),
               drive(C)])
```

The semantic/pragmatic interpretation of this question is done using the ontology of concepts and it allows to obtain the following DRS:

```
drs(entities:[A:(who,male/fem,sing),
              B:(indef,male/fem,sing/plu),
              C:(def,fem,sing),
    conditions:[homicide(A,B),
               person(A),
               person(B),
               drive(A,_,_,_,C),
               alcohol\_rate(C),C>0.5])
```

- In order to perform the first step of the answer generation the following approach is followed:
  We keep the referent variables of the question and we try to prove the conditions of the DRS in the knowledge base. If the conditions can be satisfied in the knowledge base, the discourse referents are unified with the identifiers (skolem constants) of the individuals.
- The next step is to retrieve the words that constitute the answer:

In this phase we should retrieve the conditions about the identified referent A and choose which ones better characterize the entity. Our first option is to choose a condition with the predicate *name* (name(A,Name)).
However, it is not always simple to find the adequate answer to a question. See, for instance, the following questions:

∗ How many times was Mr. X accused?
∗ What crimes committed X?
∗ How were Mr. X crimes committed?

In order to choose the best answer to a question our system has an algorithm which takes into account the syntactical category of the words that may appear in the answer and it tries to avoid answers with words that appear in the question.
Questions about places or dates have a special treatment involving the access to a database of places or dates.

Note that several answers may exist for a specific question. We calculate all possible answers and present them to the investigator.

Our system is able to obtain answers with conditions in several documents but, at this phase, we constrained the system to obtain only answers with referents introduced in the same sentence (predicate *referred_in* allows to obtain that information –see page 7).

8. Evaluation

At present we do not have final results for the evaluation of our QA-system applied to the juridical domain. We only have results for a general domain – Portuguese newspapers – in the context of CLEF – Cross Language Evaluation Forum – 2005. However, preliminary results for the juridical domain seem to be similar with the ones obtained at CLEF. In this forum a set (200) of questions was elaborated by a jury and given to the system. The system's answers were, then, evaluated by the same jury.

Our system had the following results:

- 25% of correct answers (50 answers).
- 1.5% correct but unsupported answers (3 answers).
- 11% inexact answers – too many (or too few) words (22 answers).
- 62.5% wrong answers (125 answers).

The system had 125 wrong answers, but it is important to point out that 105 of these wrong answers were NIL answers, i.e., situations were the system was not able to find any answer to the questions. So, only in 10% of the situations (20 answers) our system gave a really wrong answer.

A preliminary analysis of the other incorrect answers showed that the main cause of problems in our system is related with lack of knowledge: wrong syntactical analysis; lack of synonyms; and, mostly, an incomplete ontology. In fact, most problems are related with incorrect pragmatic analysis due to an incomplete ontology.

This problem analysis is valid also for the juridical domain. It is crucial for the performance of the QA-system to have a good representation of the domain; a good ontology is the key-factor in this process.

9. Conclusions and Future Work

We propose an architecture for a question answering system for the Portuguese language and we applied it to the legal domain.

Our system uses natural language processing techniques to create a knowledge base with the information conveyed by documents. Queries are analysed by the same tools and logical inferences over the knowledge base are performed, trying to find an adequated answer. The inference process is performed using a logic programming framework and the Prolog inference engine.

The system main problems are related with errors in the several NLP tools and with the lack of coverage of the ontology.

As future work, we intend to explore the problem of automatically build ontologies. The improvement of the used NLP tools is another area needing much work: to improve the syntactical parser and, specially, the semantic analyser (which is a quite open problem in the NLP community). We also intend to handle anaphoric relations in the documents, allowing the reduction of the number of distinct referents.

References

[1] Salvador Abreu. Isco: A practical language for heterogeneous information system construction. In *Proceedings of INAP'01*, Tokyo, Japan, October 2001. INAP.

[2] Salvador Abreu, Paulo Quaresma, Luis Quintano, and Irene Rodrigues. A dialogue manager for accessing databases. In *13th European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 213–224, Kitakyushu, Japan, June 2003. Kyushu Institute of Technology. To be published by IOS Press.

[3] Eckhard Bick. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000.

[4] Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as abduction. Technical Report SRI Technical Note 499, 333 Ravenswood Ave., Menlo Park, CA 94025, 1990.

[5] Hans Kamp and Uwe Reyle. *From Discourse to Logic:An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: D. Reidel, 1993.

[6] Paulo Quaresma and Irene Rodrigues. Using dialogues to access semantic knowledge in a web legal IR system. In Marie-Francine Moens, editor, *Procs. of the Workshop on Question Answering for Interrogating Legal Documents of JURIX'03 – The 16th Annual Conference on Legal Knowledge and Information Systems*, Utrecht, Netherlands, December 2003. Utrecht University.

[7] José Saias. Uma metodologia para a construção automática de ontologias e a sua aplicação em sistemas de recuperação de informação – a methodology for the automatic creation of ontologies and its application in information retrieval systems. Master's thesis, University of Évora, Portugal, 2003. In Portuguese.

[8] José Saias and Paulo Quaresma. Using nlp techniques to create legal ontologies in a logic programming based web information retrieval system. In *Workshop on Legal Ontologies and Web based legal information management of the 9th International Conference on Artificial Intelligence and Law*, Edinburgh, Scotland, June 2003.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

How Technology Can Help Reducing the Legal Burden

Tom van Engers[a], Ron van Gog[b] and Arian Jacobs[c]
[a]*Leibniz Center for Law, University of Amsterdam,*
[b]*Belastingdienst, The Netherlands,*
[c]*Rulewise, The Netherlands.*

Abstract. This short paper explains how relative simple technology can help governments to reduce the legal burden.

1. Introduction

The Dutch Tax and Customs Administration (DTCA) as any other public administration publishes decisions in how to interpret the discretionary room that always is present in legislation in the form of policy directives which have a legal status. As part of a national programme aiming at reducing the administrative burden and reducing (perceived) regulatory pressure (see also [1]), the DTCA started a task force aiming at reducing the number of these policy directives from about 1400 policy directives to about half. We supported this project with technical solutions that help the project manager and her crew to intelligently prune the set of policy directives and consequently meet her project's goals. The technology used wasn't exactly rocket science, consequently we don't claim that we present highly innovative AI-based techniques, but we want to show a useful application of AI-based techniques in a practical legal setting.

2. The Clustering Process

Policy directives are a solution to the unintended openness of legal concepts and serve both in diminishing discretionary room for civil servants and improving transparency and equality before the law for the citizen. Through the years many of such policy directives have been published. It is both in the interest of the citizens (taxpayers) as that of the civil servant themselves that the number of such policy directives doesn't become too large.

For clustering we first created a vector representation of the policy directives. Every dimension is associated with a word (normally from a list of keywords) in the source document. The number of dimensions is thus determined by the number of words n. Every coordinate is determined by the weight of the corresponding words (dimensions). The weight of a word w in document I is calculated as:

$$weight(w,i) = f_{wi} \log(\frac{N}{n_w})$$

In this formula f_{wi} is the frequency in which the word w occurs in document i, N the number of documents in the corpus and n_w the number of documents that contain the word w. This way the weight of a word (in a single document) is proportional to the number of occurrences in the document. The factor $\log(N/n_w)$ causes that the weight is weakened proportionally to the occurrences in other documents in the corpus. A word occurring in all documents consequently will have a weight 0 in all documents (no significance). We normalised the vectors in order to make them comparable. Meaningless words such as determiners or propositions, are included as dimension but will be factored out by $\log(N/n_w)$.

The correspondence between documents can be expressed as the dot product of the documents' vectors. The dot product of two normalised vectors is equal to the cosine of the angle between those vectors. A value 1 means that the angle between the vectors is equal to 0 degrees which implies that the documents have (almost) the same meaning.

We clustered the policy directives (using 0,7 as cut-off value) which resulted in 380 clusters with an average size of 2.74 documents and a maximum cluster size of 58. We then presented the clusters to the experts. These experts were surprised that we were able to offer them a set of correlated documents without having any legal knowledge about the complex legal topic that was described within them. We even included documents that they overlooked, but were relevant.

3. Conclusions

The project mentioned in this short paper is just an example of an attempt to reduce the legal burden. The approach explained in this paper proves that in order to support legal experts it is not always required to use rocket science. Using existing technology sometimes with little extensions or used in a slightly different way can already provide a lot of help in difficult legal tasks. The full paper can be found on www.LeibnizCenter.org.

References

[1] Arendsen, R. and Van Engers, T.M., An Empirical Study on Law Enforcement and Data Reporting Strategies, in: Proceedings of E-Government Workshop in conjunction with JURIX 2003, Palmirani, M. et al., Universitatsverlag Rudolf Trauner, December 2003.

[2] Van Engers and T.M., Legal engineering: A knowledge engineering approach to improving legal quality. In J. Padget, R. Neira, and J.L. De León, editors, *eGovernment and eDemocracy: Progress and Challenges*, pages 189-206. Instituto Politéchnico Nacional Centro de Investigacion en Computación, 2004. ISBN 970-36 0152 9.

[3] Van Engers, T.M. and Boekenoogen, M.R., Improving legal quality - a knowledge engineering approach. In Galindo, editor, *International Review of Law Computers & Technology*, volume 18. Carfax Publishing, March 2004. ISSN 1360-0869.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

A Semantics-based Visual Framework for Planning a New Bill

Carlo Biagioli and Enrico Francesconi [1]
ITTIG-CNR, v. Panciatichi 56/16, 50127, Florence, Italy

Abstract. In this paper a module able to guide the legislative drafter in planning a new bill is presented. This module aims at helping the legislative drafter to build a new act from a conceptual point of view. Using this module the classical drafting process is inverted: the structure of a bill is constructed on the basis of its semantics.

Keywords. Provision Model, OWL, Planning, Semantic Legal Drafting.

1. Introduction

This project aims at building a module for planning new bills. It is conceived as a part of a project to implement a "law making environment" for the production and reviews of legislative texts. NIREditor [1] is the central element of this environment, being a law drafting module able to produce law texts according to the standards established by the "Norme in Rete" (NIR) project. It is based on a double vision of the legislative text: formal and functional [2]. Articles, paragraphs, etc., are entities of the formal profile, while functional entities are *provisions* (ex. *definition*, *obligation*, *sanction*, *amendments*, etc.) and their *arguments* (ex. the *addressee* of an obligation) represented as metadata in the DTD NIR. The planning module presented in this paper aims at providing the legislative drafter with facilities for the construction of a new bill starting from the definition of its semantics in terms of provisions and their arguments.

2. The Planning Module

This project aims at providing users with facilities to help the organization of a new legislative text according to a semantic point of view. In the traditional legal drafting the formal structure of a document may not be the best one to express the semantics of the text. The Planning module aims at turning over the traditional legal drafting process, providing facilities to firstly express the semantics (functional profile), then to organize the semantic components in a well-suited formal structure. In this process metadata are chosen by the legislative drafter himself: therefore they are authentic metadata [3], as chosen by the legislative drafter, and they are used also to guide the drafting phase.

The planning activity basically consists in describing how the domain of interest of the human activities (a scenario) will be regulated by the new act. The formalization of this can rely on a model of the scenario to be regulated (terms and concepts drawn from or organized into term hierarchies (thesauri) and concept taxonomies (ontologies)) and on

[1]Correspondence to: Enrico Francesconi, ITTIG-CNR, via Panciatichi 56/16, 50127 Florence, Italy, Tel: +39-055-43995, Fax: +39-055-410644 email: francesconi@ittig.cnr.it

a model of the possible ways the act will regulate the scenario (model of *provisions* and *arguments*). The established relations and the instances of these models will represent the semantics (the functional profile) of the bill under construction. Facilities will be provided helping the organization of the semantic components (provisions) into formal partitions of the constructing act. At the end of this process, the formal partitions of the act will contain semantically correlated components (provisions), and the semantically qualified formal structure skeleton of the new act can be obtained. Partition wording can rely upon the user, or proposals of partitions wording can be generated on the basis of the semantics of the provision associated to each partition.

3. The Planning Module Software Architecture

The Planning module is conceived as a visual editor of provisions and it is composed by four main elements: a Model of Provisions and Arguments, a Model of the scenario to be regulated, a Definition manager, a Visual Provision Manager.

The Model of Provisions and Arguments is represented using OWL, since it guarantees interoperability among applications relying on it and a well-grounded framework for reasoning on provisions. The Model of the scenario is described by legal concepts or terms drawn from JurWordNet or from ontologies and lexicons of the domain of interest: usually they are the description of the involved actors (norm addressees), the main activities, the regulated actions, and in general the entities of the domain regulated by the law. Along with possible new terms, they can be collected in a Definition Manager. They will be used mainly as provision argument values (ex: arguments of *Definitions*). The Visual Provision Manager is a visual panel where provisions types and their arguments can be inserted and handled as visual objects, thus defining the functional profile of the new act. In a second phase the user will be able to provide the best structural organization (formal profile) of the text, with the advantage to see the building blocks ("paragraphs") of the bill under construction from the point of view of their meaning ("provisions"). Provision instances can be grouped, through queries on provisions and arguments, into semantically correlated clusters ("articles", from the formal point of view). This process can be recursively repeated, obtaining a tree representing the semantically annotated document formal profile, to be worded (manually or semi-automatically) being guided by the semantic annotation.

4. Conclusions

A module able to guide the legislative drafter in planning a new bill from a conceptual point of view has been presented. The traditional drafting process is inverted: the structure of a bill (formal profile) is constructed on the basis of its semantics (functional profile), so to obtain a well-structured text, where the chosen formal structure fits well the functional profile. A prototype is currently under implementation on a Java platform.

References

[1] C. Biagioli, E. Francesconi, P. Spinosa and M. Taddei, "A legal drafting environment based on formal and semantic XML standards", *In Proceedings of the Tenth International Conference on Artifi cial Intelligence and Law*, pp. 244–245, 2005.

[2] L. Branting, J. Lester and C. Callaway, "Automated Drafting of Self-Explainig Documents", *In Proc. of the Sixth Int. Conference on Artifi cial Intelligence and Law*, pp. 72–81, 1997.

[3] F. Lachmayer and H. Hoffmann, "From legal categories towards legal ontologies", *In Proc. of the Workshop on Legal Ontologies and Artifi cial Intelligence Techniques*, pp. 63–69, 2005

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

The Desocialization of the Courts, Sentencing Decision Support and Plea Bargaining

Andrew Vincent[a,b] and John Zeleznikow[a]
[a]School of Information Systems and International Corporate Governance Research Institute, Victoria University of Technology [b]Greek Studies, La Trobe University, Australia.
andrew.vincent1@research.vu.edu.au

Judicial decision making in the legal domain is a complex task involving the evaluation large amounts of information, analysing issues, legislation, and precedent as well as assessing feasible options. One of the central and perennial questions of sentencing law and scholarship is how law-makers should pursue an appropriate balance between consistency and individualisation in punishment. In Victoria (Australia) legislatures continue to grant judges considerable discretion in sentencing as distinct from some jurisdictions in America. In these jurisdictions, parliament has, arguably, favoured individualisation over consistency. Even these jurisdictions it is desirable that like cases be treated alike. From a retributivist perspective, a certain measure of consistency is necessary to ensure that offenders are punished in (at least rough proportion) to their culpability and thereby maintain public confidence in the integrity of the criminal justice process. From an economic/utilitarian perspective, consistency enhances certainty of punishment which increases law-makers' ability to pursue optimal levels of deterrence.

Victoria Legal Aid (VLA) is the principal provider of legal aid in Victoria and the largest criminal law practice in the state handling upwards of 80% of criminal law defense cases. It is thus in VLA's interests to be able to provide sound advice regarding possible sentences especially as a result of guilty pleas. To support this goal we have developed a decision support system to provide advice about possible sentences, so that VLA defense lawyers can make informed decisions about how to best conduct plea bargaining. Most sentencing decision support systems provide statistical sentencing information for judges. Our system is unique in focusing upon the needs of defence counsel.

The Magistrates Court has a surfeit of commonplace cases. These cases can be used to model how magistrates exercise discretion in deciding sentences. There is a growing trend in the criminal justice system to expedite the movement of cases through the system towards final disposition. In Australia plea bargaining and summary disposition of cases are used to accelerate the criminal justice process. This process is responsible for the desocialization of the courts. Desocialization is facilitated by the removal of social information.

It has been suggested that the process of desocialization of the law, which primarily involves the reduction of social information available to all persons with discretionary power, reduces levels of discrimination. Legal variation is a direct function of social information. Social information is best understood as information that locates individuals within the society they inhabit: racial, economic, cultural, organisational and so on. As the amount of social information increases the more variation is evident, especially with respect to the criminal case. Social information equates to the aggravating and mitigating factors that under the Victorian sentencing regime are to be taken into account when sentencing a criminal. Some social information, for example, that which may be inferred by physical appearance is not to be taken in to account by the judiciary. The modeling that has been conducted (in consultation with domain experts) for the sentencing project, has identified relevant factors that judges and magistrates regularly take into account when sentencing convicted criminals.

The suggestion that legal variation is a direct function of social information means that consistency of outcomes in the sentencing process can be achieved by the removal of all information that places an offender in a social context. Thus consistency of sentence would be achieved based solely on the technical aspects of the case. Sentencing guidelines in the United States require judges to determine the numerical levels associated with the offence of conviction, specific characteristics of the offence (for example, the use of weapon) and the offender’s criminal history value. The calculation then reveals an offence level and criminal history category, a sentence is then derived by reference to a sentencing grid. When the Minnesota sentencing guidelines were first established, one of the main purposes was preventing racial, gender, and socioeconomic disparities with respect to sentence. Only a very small amount of social information is allowed to enter the sentencing equation under the United States style of guideline sentencing.

The sentencing decision support system is only the first element of a larger online plea negotiation environment to facilitate negotiation between prosecution and defence (VLA) lawyers. The sentencing system provides a BATNA (Best Alternative to a Negotiated Agreement) for a plea bargain. In an attempt to facilitate the development of online dispute resolution environments [1] have suggested a three-step model combining dialogue tools and negotiation support systems. It is anticipated that the Lodder-Zeleznikow framework can be used profitably to assist in the development of this system by: i). using the sentencing decision support system to provide advice about possible BATNAs; ii). developing a process that enables direct communication and negotiation between the parties which supports interest based communication; iii). Developing a process that provides negotiation support through the use of compensation and trade-off strategies.

The sentencing decision support system provides a way of structuring social information so that a sentence indication can be provided and an informed decision about a plea can be considered.

References

[1] Lodder, A. and Zeleznikow, J., Developing an online dispute resolution environment: Dialogue tools and negotiation systems in at three step model. *The Harvard Negotiation Law Review*,10, 287-338, 2005.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Towards Modeling Systematic Interpretation of Codified Law

Matthias Grabmair [a] and Kevin D. Ashley [b]
[a] *University of Augsburg School of Law, Germany, Matthias.Grabmair@gmx.de*
[b] *School of Law, Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA, +1 (314) 624-7496, ashley+@pitt.edu*

Abstract. This short paper introduces the *Interaction Predicate* model, which attempts to model some aspects of systematic interpretation of codified law. It introduces an intermediate rule representation containing dynamic reasoning elements which make use of domain knowledge ontologies.

Keywords. Systematic Interpretation, Legislation, Rule Based Reasoning

1. Legal Theory Foundation and Objective

Rule based reasoning in AI&Law faces the task of construing determinate rules from sometimes vague and ambiguous legal sources. Our *Interaction Predicate* model (IP) is intended to approach this issue by emulating techniques of systematic interpretation that take account of the influence of the code's structure and system on the legal meaning of its provisions. Systematic interpretation in civil law theory distinguishes between two categories of codified statements. *Complete* legal sentences provide enough information by themselves to be translated into an unambiguous IF-THEN relation as it is needed for a proper application of a rule to a case. *Incomplete* legal sentences need to be read in conjunction with other legal sentences in order to compose such an autonomous regulation. Moreover, a code's systematic structure can be divided into the *exterior*, being the text's sections, subsections etc., and the *interior* structure, mainly focusing on the substantive correspondence of the terminology. The current objective of the IP concept is to formally model this systematic interpretation method and, later on, to develop an experimental environment which is supposed to reason with examples of codified law. It can also visualize each norm's *sphere of influence* inside the system, that is the set of norms with which a certain norm interacts directly as well as through other norms, and thereby show its impact on the code as a whole.

2. Explanation of the Interaction Predicate Approach

To allow a more accurate reflection of the code, the model uses common IF-THEN relations with binary predicates to represent each norm, but enriches them with dynamic elements, so-called *Interaction Predicates*. These symbolize standard phenomena of norm

interaction and have reasoning algorithms attached to them, which are defined as the typical thinking steps a jurist takes when systematically interpreting the respective norm. In this intermediate norm representation (INR), every legal sentence of the original code has its own relation, so that the *exterior* system of the code is preserved. The *interior* system is captured by a set of domain knowledge ontologies, which interrelate the binary predicates and embed them in the overall legal construct. In the actual reasoning process, this INR is translated into a tree of unambiguous IF-THEN rulesets, meaning rules which are free of interaction predicates. The IP reasoning algorithms use the domain ontologies to resolve interpretive issues and, whenever more than one legally plausible interpretation is determined, spread open the tree's branches accordingly.

For example, a general norm *n1* may order a legal consequence under a set of factual requirements *r1*, *r2* and *r3*. A subsequent norm *n2* provides that requirement *r3* alone "shall suffice" in case a fourth requirement *r4* is present, but neither states *n1*'s consequence anew nor makes an explicit reference to it. The IP model will represent *n2* by using the interaction predicate *suffice* as its consequence and enable it to determine *n1* as the referenced norm by virtue of *n2*'s structural position relative to *n1*. It will take over *n1*'s consequence for *n2* and, by using domain knowledge, determine whether *r3* and *r4* really make both *r1* and *r2* obsolete or whether there is an implicit assumption that only one of them shall be disregarded while the other one remains required.

3. Related Research

The IP concept assumes that a piece of written law can principally be translated into a normalized INR set. *De Maat* and *Van Engers* are already developing an automated rule extraction model (*de Maat & Van Engers* [1]). The POWER project (*Van Engers et. al.* [2]) included an anomaly detection framework which can spot inconsistencies in modeled legislation and prompt an expert for reviewal. By contrast, the IP model is conceptualized to resolve norm interaction conflicts by utilizing formalized legal methods.

4. Concluding Remarks

The *Interaction Predicate* approach tries to capture civil law legal reasoning more accurately by introducing an INR which neutrally preserves the *sphere of influence* of each norm in the context of the code and underlying domain knowledge. Its functionality depends on the definition of the IP algorithms, which may unnecessarily complicate undisputed issues. Still, we are convinced that experimental results can help sharpen the contours of legal methodology and contribute to norm interpretation research in AI&Law.

References

[1] E. de Maat & T. van Engers, *Mission Impossible?: Automated Norm Analysis Of Legal Texts*, Jurix 2003: The Sixteenth Annual Conference - Legal Knowledge and Information Systems, 143-144 (D. Bourcier ed., IOS Press 2003).

[2] T. van Engers et al., *POWER: Programme for an Ontology based Working Environment for Modeling and Use of Regulations and Legislation*, Proc. of the 11th Workshop on Databases and Expert Systems Applications (IEEE) 327-334 (Tjoa, Wagner, and Al-Zobaidie ed., 2000).

Large-Scale Linguistic Ontology as a Basis for Text Categorization of Legislative Documents

Natalia Loukachevitch [a,b], Boris Dobrov [a,b]
[a] *Research Computing Center of Moscow State University (NIVC MSU), Russia*
[b] *NCO Center for Information Research, Moscow, Russia*

Abstract. The paper describes the structure and properties of a large linguistic ontology – a new kind of information retrieval thesaurus - Thesaurus on Sociopolitical Life for Conceptual Indexing. The thesaurus is used in various real-scale information-retrieval applications in the legal domain. At present one of the main applications of the Thesaurus is knowledge-based text categorization. Categories are connected with the Thesaurus by flexible relationships. The categorization system can process text collections containing texts different in sizes and types.

Keywords. Linguistic Ontology, Information Retrieval Thesaurus, Text Categorization, Legal Documents.

Governmental and legislative bodies of various countries often need their documents to be classified according to large and complicated hierarchical systems of subject headings. As a rule, collections of legislative documents are very large, the documents differ greatly in sizes and styles. Manual classification of such collections is hard, time-consuming and tends to subjectivity, which can increase because of lack of qualified specialists. Therefore use of automatic methods of text categorization in the domain is very important.

Use of machine-learning techniques for text categorization of legal documents is hampered by such factors as serious inconsistency of training data, lack of enough training documents for each subject heading, poor explanation means of existing machine learning approaches.

On the other hand knowledge-based methods [1] require huge volumes of knowledge described, because normative documents regulate relations in various domains: state policy, governmental bodies, taxes, accounting, banking, industrial and agricultural production, housing and social welfare, nature protection, science, education, defense and many others. Terms from all these domains can be essential for the document processing.

In fact to process normative documents it is necessary to have a knowledge resource comprising “descriptions” of different situations and problems of the contemporary society life.

We call this polythematic domain “sociopolitical domain” and for more than ten years develop a linguistic resource called Thesaurus on Sociopolitical Life for Conceptual Indexing (or Sociopolitical thesaurus) [2].

Now the thesaurus is a hierarchical net of concepts comprising more than 32 thousand concepts, 79 thousand Russian terms, 80 thousand English terms. In construction of the thesaurus we combined three different methodologies:

- The methods of construction of **information-retrieval thesauri** (information-retrieval context, analysis of terminology, terminology-based concepts, a small set of relation types)
- The development of **wordnets** for various languages (word-based concepts, detailed sets of synonyms, description of ambiguous text expressions)
- Ontology and **formal ontology** research (strictness of relations description, necessity of many-step inference).

These features allow us consider our thesaurus as a **linguistic ontology**.

For development of a text-categorization system the categories are described as Boolean formulas using a relatively small number of ‘supporting’ concepts of the Sociopolitical thesaurus. The formulas can be expanded on the basis of properties of the thesaurus relations.

Possibility to process texts of various types and size is based on the thematic representation of the text contents, where the terms of a text are divided to thematic nodes, simulating elements of the main theme and the subthemes of a text. Construction of the thematic representation is based on such a property of texts as lexical cohesion.

Concept-based automatic categorization gives opportunity to explain received results to legal experts.

In the legal domain we have developed two text-categorization systems: for governmental bodies and for a commercial organization.

The corporate subject headings system included more than 3000 subject headings. As training materials they provided more than 250 thousand legal documents (federal, regional, subordinate documents, court practice) with manually established subject headings.

Performance results of the text categorization system were as follows: 75% recall, 20% precision. During our work we revealed that these performance figures estimate not only the quality of the automatic categorization but also the quality of the manual categorization, its inconsistency and low recall. The concept-based approach to representation of categories allowed us to demonstrate these problems to experts of the company who began to seek ways to improve the situation.

References

[1] Ph. Hayes, Intelligent High-volume processing using shallow, domain-specific techniques, *In: Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* New Jersey, pp.227-242, 1992.

[2] N. Loukachevitch, B. Dobrov, Evaluation of thesaurus on sociopolitical life as information retrieval tool, *In: “Proceedings of Third International Conference on Language Resources and Evaluation (LREC2002)”*, M.Gonzalez Rodriguez, C. Paz Suarez Araujo, eds. – Vol.1 – Gran Canaria, Spain, pp.115—121, 2002.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Game Mechanisms & Procedural Fairness

Moshe Looks [1] and Ronald P. Loui
Dept. of Computer Science, Washington University in St. Louis, USA

The main paper to which this abstract refers models procedural fairness and procedural justice, crucial concepts in the design and appraisal of social interaction [1]. Argumentation systems in particular owe the quality of their inferences to the quality of the procedure that produces them. We are interested in both games that can be justified without reference to substantive social purposes, such as tournaments, and games that are fair, even if they are formally asymmetric, because they are appropriate to their social purposes, such as litigation and prosecution games.

A mathematical theory of procedural fairness has eluded theorists (much informal discussion in this regard has centered around the work of Rawls [2]). Such a theory would give formal standing to devices such as anonymization through chance and turn-swapping, equality of opportunity, symmetry of rules, and exchangeable asymmetries. It would understand the fundamental role of fair procedure: to construct justifiable ex post asymmetries that could not be justified except by reference to the procedure that produced them. Such a procedure begins with a justifiable ex ante position, and constructs its outcome on serendipity of play and chance, under an independently justifiable regimen.

We have developed a preliminary formal framework for exploring some mathematical properties of procedural fairness. The framework encompasses deterministic and stochastic games and player strategies, and contains formal devices for recursively composing complex games from simple components, including devices concerned with modifying/introducing procedural fairness properties. We also define allocation games as a special case, and may then invoke familiar game-theoretic concepts. Consider

$$jkp^{+}(m_0, m_1 \in \{Jan, Ken, Po\}) = \begin{cases} jkp^{+} & \text{if } m_0 = m_1 \\ 0 & \text{if } (m_0, m_1) \in wins \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

a partial definition of *Jan-Ken-Po* (Rock-Paper-Scissors).

Chance mechanisms, in the form stochastic outcomes or symmetrical moves, may substitute for arbitrary choices and introduce fairness. For example, one may define, with our framework, a generic device for transforming a deterministic, asymmetrical game (such as chess), into a stochastic, symmetrical game (i.e., by flipping a coin to determine the assignment of player roles). Chance may also be used as a progress mechanism for states with indeterminable outcomes (e.g., resolving deadlock by killing a random process). Of course, there are also situations where chance is undesirable because it creates unrepresentative outcomes. Often, this may be eliminated by *repetition*. For allocation games, this is best understood by looking at the spread of outcomes, which typically becomes "smoother" as a game is repeated.

[1] Correspondence to: Moshe Looks, Dept. of Computer Science, Washington University, St. Louis, MO 63130, USA Tel.: +1 202 641 2157; mlooks@cse.wustl.edu. Also loui@cse.wustl.edu.

Consider procedures designed to establish some social predicate such as guilt. It is constructive to distinguish between two kinds of fairness: *teleological fairness* (appropriateness, effectiveness) is concerned with the correspondence between what is measured by a procedure and what we intend it to; in contrast, *structural fairness* is concerned with internal aspects of procedures, such as arbitrariness or imbalance between players (if an even contest is intended). Time and spread are relevant to teleology; while repetition can decrease the risk of unrepresentative outcomes, it may introduce a wider spread, while devices such as “best of n” games are time-consuming, and can limit perceived decisiveness.

In addressing physical or mental player limitations, there may be trade-offs between the conflicting goals of teleological and structural fairness. For example, in a “best of n games” match, increasing n improves structural fairness, while teleological fairness is decreased, as the match eventually degenerates into a measure of stamina rather than skill. Additionally, player limitations have a decisive impact of the teleological fairness or lack thereof of a given game. While tic-tac-toe is teleologically unfair for adults, it may be appropriate for young children.

Within our framework, *dominant strategies* may be defined absolutely, with respect to a particular population of strategies, or within a particular population of strategies. Tic-tac-toe has a dominant strategy, for example, but not within the population of strategies accessible to very young children.

Another important tool for reasoning about the procedural fairness properties of games is *reducibility*, which may be formally defined in terms of mappings between games. Interestingly, many actual games that at first glance appear reducible are not; reducibility depends on the population of players. For example, actual rock-paper-scissors matches may exhibit uncharacteristically long runs of ties, and are hence irreducible along simple lines. Note that the passage of time may be ignored by some reductions, while it in fact is the purpose of some actual games.

In conclusion, our preliminary work has hinted at a framework in which procedural devices may be isolated, and fairness claims can be derived. Much work remains to be done. Thus far, we have identified structural properties of chance, termination, spread, and rule symmetry, and commented on some of the relations between them in the appraisal of procedural fairness. As society is essentially built upon a foundation of procedural fairness, and a mathematical framework permits critical appraisal, we view this work to be of value.

References

[1] Michael D. Bayles. *Procedural Justice: Allocating to Individuals*. Kluwer Academic, New York, NY, 1990.

[2] J. Rawls. *A Theory of Justice*. Belknap Press, Cambridge, MA, 1999.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Practical Reasoning and Proposing: Tools for e-Democracy

Douglas Walton
Department of Philosophy, University of Winnipeg, Canada

Abstract. It is shown how two tools developed in argumentation theory are useful for AI systems for electronic democracy [2,3] and more generally for formal dialogue systems representing deliberation. The novel part of this analysis is that it represents the speech of proposing as a small dialogue exchange in which one party practically reasons with another, based on premises that both are committed to, as collaborative participants in a deliberation dialogue. The structure of practical reasoning as a type of argument as analyzed in [6] is brought to bear, to bring out special features of the speech act of proposing that make it a nice fit with the formal framework for deliberation dialogue constructed by [4].

Key Words. Argumentation, Formal dialogue systems, Deliberation, Electronic democracy, Argumentation schemes, Speech acts

Three dialectical adequacy conditions are formulated, based on the accounts of proposing in the argumentation literature [1,5].

1. Dialectical Adequacy Conditions for Defining the Speech Act of Making a Proposal

Condition 1. In making a proposal, the proponent puts forward a statement (proposition) that describes an action and says that both proponent and respondent (or the respondent group, in a greater than 2 multi-agent case) should carry out this action. The putting forward of this statement is done in such a way that it commits the proponent to carrying out that action, and urges the respondent to take on commitment to it as well. The statement has the logical form of the conclusion of a practical inference. It is often described in the literature as a practical ought-statement. Thus the statement contains an action, and also expresses a kind of attitude toward that statement, saying essentially, “We ought to do it”.

Condition 2. The proponent puts forward the statement with the aim of offering reasons of a kind that will lead the respondent to become committed to it, either now or at some later point in the dialogue. These reasons are practical in nature. The premises that provide the reasons are directed to the goals of the respondent, and to actions that can be accepted by the hearer as means to attaining these goals. The inference link between these premises and the conclusion (the statement to be proved by the proponent) is the argumentation scheme for practical reasoning.

Condition 3. As condition 2 makes clear, the proponent has essentially put forward an argument. As with putting forward any other argument in a type of dialogue like a deliberation or critical discussion, there is presupposed some initial doubt or conflict of opinions, and the job of the proponent is to overcome that doubt, while the job of the respondent is to express it. Thus the role of the respondent is to ask questions that cast the prudential reasonableness of the action in the statement into doubt, and to mount attacks (counter-arguments and rebuttals) against it.

Dialogue preconditions, defining conditions (offering a formal definition of the speech act), and post conditions are presented. According to the formal definition of the speech act of making a proposal, two parties are engaged in a deliberation dialogue, the proponent who puts the proposal forward, and the respondent, representing the agent or group of agents the proposal is directed to in the deliberation context. What is proposed is a proposition. According to the definition, the proponent proposes the proposition to the respondent if and only if (1) there is a set of premises that the proponent is committed to, that fit the premises of the argumentation scheme for practical reasoning (2) the proponent is advocating these premises, that is, he is making a claim that they are true or applicable in the case at issue, (3) there is an inference from these premises fitting the argumentation scheme for practical reasoning, and (4) the proposition is the conclusion of the inference. This definition performs the interpreter function in any formal dialogue system. It checks whether any move made in the dialogue fits a defined speech act or type of move that is recognized as legitimate.

References

[1] Aakhus, Mark, 'The Act and Activity of Proposing in Deliberation', paper presented at the ALTA Conference, Alta, Utah, August, 2005.

[2] Atkinson, Katie; Bench-Capon, Trevor and McBurney, Peter 'Justifying Practical Reasoning', *Proceedings of the Fourth Workshop on Computational Models of Natural Argument* (CMNA 2004), ECAI 2004, Valencia, Spain, pp. 87-90.

[3] Atkinson, Katie; Bench-Capon, Trevor and McBurney, Peter 'PARMENIDES: Facilitating Democratic Debate', *Electronic Government*, ed. R. Traunmuller, Lecture Notes in Computer Science (LNCS), 3183. Third International Conference on eGovernment (EGOV 2004), DEXA 2004, Zaragoza, Spain, 2004a.

[4] Hitchcock, David; McBurney, Peter and Parsons, Simon 'A Framework for Deliberation Dialogues', *Argument and Its Applications: Proceedings of the Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (OSSA 2001),* eds H. V. Hansen, C. W. Tindale, J. A. Blair and R. H. Johnson, 2001. Also available on Peter McBurney's web page (2005) at the University of Liverpool, Department of Computer Science: http://www.csc.liv.ac.uk/~peter/

[5] Kauffeld, Fred J. 'Presumptions and the Distribution of Argumentative Burdens in Acts of Proposing and Accusing', *Argumentation*, 12, 1998, 245-266.

[6] Walton, Douglas *Practical Reasoning: Goal-Driven, Knowledge-Based, Action-Guiding Argumentation*, Savage, Maryland, Rowman & Littlefield, 1990.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

The Legal Concepts and the Layman's Terms

Bridging the Gap through Ontology-Based Reasoning about Liability

Ronny van Laarschot [a], Wouter van Steenbergen [b], Heiner Stuckenschmidt [a], Arno R. Lodder [b] and Frank van Harmelen [a]

[a] *Department of Artificial Intelligence, Vrije Universiteit Amsterdam*
[b] *Computer/Law Institute, Vrije Universiteit Amsterdam*

Abstract. The aim of the BEST-project is to support laymen in judging their legal position through intelligent disclosure of case-law in the area of Dutch tort law. A problem we have to face in this context is the discrepancy between the terminology laymen use to describe their case and the terminology found in legal documents. We address this problem by supporting users to describe their case in common sense terms taken from an ontology. We use logical reasoning to automatically determine law articles that are relevant for determining liability of parties in a case based on this description, thus bridging the gap between the laymen's description and the terminology relevant for certain articles that can be found in legal documents. We introduce the BEST-project and describe the ontology built for supporting case descriptions focussing on its use for automatically determining relevant articles of law.

Keywords. Legal Ontologies, Logical Reasoning, Mediation, Document Retrieval

1. Introduction

Laymen can turn to legal professionals to determine their legal position, but often resolve their disputes in an informal way, e.g. by mediation, negotiation. In the BEST-project (Batna Establishment using Semantic web Technology, see http://best-project.nl) we strive to provide disputing parties with information about their legal position in a liability case. In this way parties are given the opportunity to form a judgment about whether they could hold another party liable for certain caused damage or if they could be held liable themselves. Also, parties can determine how much room for negotiation is available when settling the damage. In particular this information is important, since disputes are most frequently settled between parties themselves, rather than in court or with support of a (legal) expert(s) [12]. In addition, this information might be used to evaluate the legal advice of the attorney or an other legal professional. Naturally, it also helps parties to decide whether it is beneficial to take their case to court. Ideally, at the beginning of the negotiations, parties have an idea of what the outcome would be if their dispute would be decided by a judge. By information about previous court decisions, where relevant taking into consideration other factors such as time, costs, emotions, etc.,

a well-rounded impression is obtained about a parties' BATNA (Best Alternative To a Negotiated Agreement), that is: the result that should ideally at least be reached in the negotiations (the threshold). The BEST-project aims to provide disputing parties in a stage before they seek professional assistance with information about their position in the negotiations, to assist them in the dispute, get information about the legal possibilities to claim compensations, etcetera. The target user group of the program will generally consist of laymen in the field of law, who want to get an insight into the legal aspects of their dispute. In this paper, we focus on the problem of bridging the gap between the description of a case as it might be produced by a layman and the legal terminology that is used in legal documents. In particular, we present an approach for automatically determining relevant law articles based on an abstract description of a legal case. The paper is organized as follows. In section 2 we review the aim and the scope of the BEST-project. In section 3 we introduce a formal ontology of Dutch tort law that has been developed in the project. The use of this ontology for determining relevant articles based on logical reasoning is presented in section 4. We conclude with a discussion of the work in section 5.

2. The BEST-Project

A concept central to the BEST-project is a BATNA [6][13]. Fisher and Ury introduced principled negotiation, which advocates separating the problem from the people. Fundamental to the concept of principled negotiation is the notion of Know your best alternative to a negotiated agreement (BATNA). The reason you negotiate with someone is to produce better results than would otherwise occur. Or, as [1] puts it:

> Settlements are truly informed and voluntary only if the parties choose them with a full understanding of their alternatives.

If you are unaware of what results you could obtain if the negotiations are unsuccessful, you run the risk of:

1. Entering into an agreement that you would be better off rejecting; or
2. Rejecting an agreement you would be better off entering into.

In their three step model Lodder and Zeleznikow set forth three basic stages for the effective resolution of online disputes [9]:

1. Determining a BATNA, which helps the disputing parties determine what will happen if the dispute is not resolved;
2. Allowing parties to communicate among themselves using dialogue techniques; and
3. Using game theory techniques that employ compensation/trade-off strategies to attempt to resolve remaining issues in dispute.

The BEST-project aims at facilitating the establishment of a BATNA (step 1), through intelligent disclosure of case-law. We do this by employing ontologies and ontology-based search and navigation, as has been developed in [10]. The legal domain we are looking at is damages disputes. Two questions are relevant here. First, damages are compensated only if the other party can be held liable. Whether this is the case depends on a number of factors such as the probability of the occurred event and the na-

ture of the damages. Once a (legal) person can be held liable, the second question is what compensation is reasonable. Both answers are relevant for determining a BATNA in damages disputes.

2.1. Aim and Approach

The BEST approach to supporting BATNA establishment is based on a number of principles that distinguishes it from other existing approaches to IT support for legal decision making. The most basic principle is that the project will provide supporting technology that prepares a legal judgement instead of trying to come up with such a judgement by itself. This preparation will consist of

- supporting the user to describe a specific legal situation
- retrieving and ranking descriptions of court decisions on similar cases

These functions will be implemented using a combination of statistical text retrieval methods and knowledge-based techniques. In particular, the idea is to ease the use of documents retrieval systems. This will be achieved by providing technological solutions for the two aspects mentioned above, in particular:

- An ontology-based interface for creating and classifying case descriptions
- An analysis component for generating search terms based on the classification of the case description

These components will be implemented and a prototype will be implemented that uses a thesaurus-based document retrieval system using a test data set. Further, the use of the components for enhancing existing search solutions for legal documents will be investigated.

The basic idea of the BEST approach is to de-couple the task of creating a meaningful and complete description of the case at hand from the task of retrieving similar cases. The rationale for this choice lies in the nature of the different terminologies used by laymen and by legal experts. An ontology suited to provide the terminology for supporting laymen in describing cases significantly differs from an ontology suited for providing the basis for annotating legal documents. This difference not only lies in the different terminology used by laymen and experts but also in the required representations. While an ontology for creating structured case descriptions needs to provide the basis for describing complex configurations of situations, the ontology for annotating legal documents will focus on the use of different words for describing the same legal concept or situation. It is easy to see that these tasks require conceptually different representations.

Besides the technical issues raised above a de-coupling of the case description and the document retrieval has several conceptual advantages:

- Depending on the user group, there can be different ways of describing cases that require different ontologies as a source of basic terms.
- Depending on the available data sources, there can be different retrieval engines that require different knowledge structures to determine relevant documents.
- The system will profit from using existing thesauri and annotations and provide added value to these systems by enriching them with query formulation support in terms of case descriptions.

- The system is able to point out potential liable parties the layman user might not have been aware of.

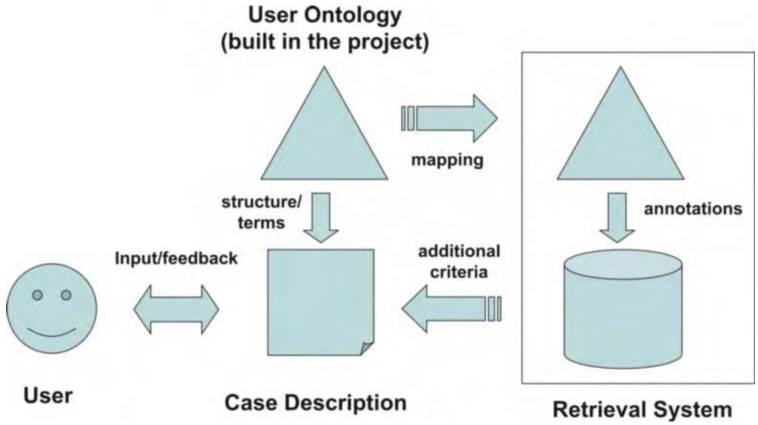

Figure 1. General Architecture of the BEST system.

Figure 1 shows a conceptual architecture that reflects the principle of de-coupling case description and document retrieval that will be the basis for the work in the BEST-project. The architecture is centered on the notion of a case description. The case description explicates relevant aspects of the case at hand using a structure and terms provided by a user ontology. This ontology is mapped on a second conceptual structure that is used to annotate legal documents.

2.2. Domain Scope

The domain we will focus upon, is that of tort law. Whereas in common law systems various torts are distinguished (e.g., nuisance, fraud, negligence, trespass to chattels), civil law systems basically know one general tort action. In the Dutch Civil Code Article 6:162 reads, as translated by Betlem [4, p 291]

> Art. 6:162 BW. 1. A person who commits an unlawful act toward another which can be imputed to him, must repair the damage which the other person suffers as a consequence thereof. 2. Except where there is a ground of justification, the following acts are deemed to be unlawful: the violation of a right, an act or omission violating a statutory duty or a rule of unwritten law pertaining to proper social conduct.

Article 162.2 clearly illustrates the general nature of the unlawful acts that qualify as torts. Dutch law defines special cases (such as liability for children, employers, objects), but most of these refer (implicitly through the notion 'fault') to this general article 162.

Basically, four types of tort typically occur more often than others, which is interesting for us since the more cases, the better the technology used in the BEST-project works, are:

- Liability in traffic (verkeersaansprakelijkheid, art. 185 WVW)
- Liability of the employer for employees who get harmed in their working place (werkgeversaansprakelijkheid voor schade door de werknemer geleden, art. 7:658 BW)

The other two fall under the general Article 162:

- Medical faults (medische fouten, art. 6:162 BW)
- Offences of violence (geweldsdelicten, art. 6:162 BW)

A reason tort law is in principle suited for the BEST-project is that compared to other areas, the rights that determine the positions of both parties can mainly be found in the Civil Code and the case law based upon it. In contract law, for instance, parties often use special contractual clauses that are not similar to clauses that were covered in previous case law. In tort law the facts may differ, but the general legal concepts applicable to facts are largely similar, and can be found in statutes and case law. Both legal sources are electronically available.

2.3. Data Sources - Changing Insights

The case law database used to disclose similar cases, is that of the public website www.rechtspraak.nl. For processing purposes we have all available 55.000 cases locally stored. Given the over 1 million legal verdicts annually, this is a low number. Nonetheless, this database contains almost all digitally available newer case law in the Netherlands. Even if commercial publishers would allow us access to their case law databases, only an additional 100.000 cases would become available electronically, most from before 1999 when the site Rechtspraak.nl was launched. We had hoped for higher numbers, and expected in particular to be able to work with a large number of cases from the lower courts. These cases are mostly not very interesting from a legal perspective, but for the BEST-project really useful. However, even the local courts do not have many cases available electronically. The relatively low number of cases available is one of the reasons that we believe it will be hard, if at all possible, to determine the amount of damages through case law retrieval. This would be different if we had as many cases as for example in the comparable Spanish project IURISERVICE [5], where in two years time over 2 million cases were collected. A possible way to partly meet this objection would be to use commercial case law databases besides the one of www.rechtspraak.nl.

Another problem is that in case law there often is no specific amount of damage compensation mentioned. The exact amount of damage usually is being determined in a separate procedure, the so-called 'schadestaatprocedure'.

All in all, to figure out the approximate amount of damage compensation that would be awarded in a court procedure will be probably done in a different way than initially thought. One option we consider is to use blind-bidding tools, such as Cybersettle.com and the like. These tools are quite popular in the insurance industry to settle cases where the only issue in dispute is the amount of damages. Beside this generic solution, for at least for one particular damage type another solution to this end is possible. With respect to injury compensation the damage compensations awarded in cases of an injury are collected in the Netherlands in the so-called 'Smartengeldgids'. We could include the information contained in this damages guide in our program to be developed. At this moment we already built an ontology based on this 'Smartengeldgids'. Damages concerned are divided into two subclasses: damage caused by an injury (injury to body or psychological damage) and damage not caused by an injury (privacy, false imprisonment etc.). All possible injuries of the first subclass are described. The domain of damages concerned with people alone consists of 189 classes.

3. An Ontology for Case Descriptions

There are three main domains integrated in the user ontology.

Tort law (hierarchy of article numbers and grounds for liability); When modeling the domain of tort law into the ontology, we mainly copied to the ontology the structure of tort law as it is described in Dutch law. The reason for this is that this structure already implies some consequences with regards to legal aspects. For instance, in Dutch tort law, there is a distinction made between direct liability and indirect liability. Further examples of liability are liability for actions of a child or possession of an animal and so forth.

Direct liability covers cases in which a person himself commits an unlawful act against another person. This unlawful act has been committed due to this persons' own fault. This is not the case when a person is liable for cases covered by indirect liability. Indirect liability is divided into liability for persons (like children or employees) and liability for objects (like animals or faulty products). When someone has a child less than fourteen years old, this person can be held liable for all damage that the child causes, whether the parent could do something about it or not.

This means that when someone describes a certain case in a user interface, the user ontology should be able to 'recognize' the various legally relevant aspects of this case. When a child of 15 causes damage, under Dutch law it is harder to hold its parents liable than when this child would have been 12 (in figure 2 we can see the different applicable article numbers).

Entities subject to law (juristic person, natural person, etc.); The division we made between natural persons and juristic persons was also made on grounds concerning legal implications. In the part where we discussed tort law, we already saw that it makes a difference if someone is a child or an adult. In the same way, it matters if an entity subject to law is a juristic person or a natural person (for example, a juristic person can never be an employee). Furthermore, in Dutch case-law it is decided that a public organization can never be regarded as a company. This means that the possibilities to hold a public organization liable are more limited than to hold a private organization liable. An overview of the division between natural persons and juristic persons, grounds for liability and the corresponding article numbers is given in figure 2.

Objects in tort law (motor vehicles, animals, product etcetera). For several objects, Dutch tort law provides a specific article concerning liability for damage caused with that object. In these different articles, different conditions apply regarding the liability. Therefore, it is important to know whether the object that caused the damage is an animal, a motor vehicle etcetera.

The ontology is completely modeled in OWL and currently contains about 300 classes most of which are actually defined in terms of logical axioms and 50 relations covering the most important aspects of the area of law introduced above as well as common sense terms for describing cases. More information about the ontology can be found in [11]. The ontology is available at http://www.best-project.nl/ontology/.

Figure 2. Partial class hierarchy of the user ontology.

4. Reasoning about Liability

A case can be described by defining individuals in the ontology: these are instances of concepts/classes and their relations or properties. For example, a horse is an instance of the class animals. The horse has the property causes with instance broken leg. Where broken leg is an instance of the class physical damage. Broken leg has the property affects with instance woman. Woman is an instance of the class plaintiff. Riding school is an instance of the class legal person. Riding school has the property owns with instance horse. The corresponding complete case description can be seen in figure 3.

The idea of the BEST approach is now to use the ontology to automatically determine the potential liability of actors in a case description. For this purpose, we extended the ontology about tort law with logical definitions of the different classes representing liability. In the following we briefly introduce the logical language used and give examples of how the logic was used to support automatic determination of potential liability. For a previous example of this approach see [7].

4.1. Description Logics

Description Logics [2] are a special type of logic that is tailored to define terminological knowledge in terms of sets of objects with common properties. Recently, description logics have become popular as a formal foundation for the Web Ontology Language

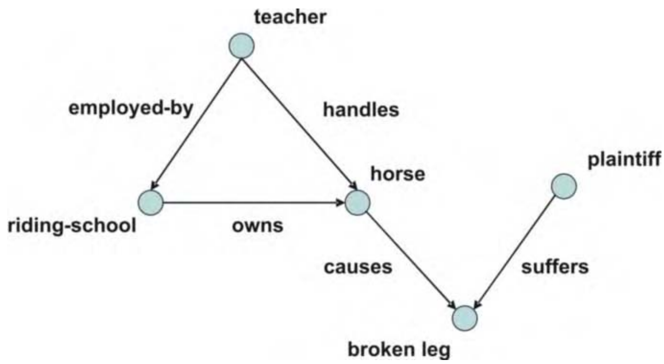

Figure 3. Case description as labeled graph.

OWL. The Basic modeling elements of a description logic are instances, concepts and relations. These modeling elements are provided with a formal semantics in terms of an abstract domain interpretation mapping $\mathcal{I}$ mapping each instance onto an element of an abstract domain Δ. Instances can be connected by binary relations defined as subsets of $\Delta \times \Delta$. Concepts are interpreted as a subset of the abstract domain Δ. Intuitively, a concept is a set of instances that share certain properties. These properties are defined in terms of concept expressions. Typical operators are the Boolean operators as well as universal and existential quantification over relations to instances in other concepts. The formal definitions can be found in the table below.

DL Expression	Semantics
A	$A^{\mathcal{I}} \subseteq \Delta$
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta - C^{\mathcal{I}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y : (x,y) \in R^{\mathcal{I}}\}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid (x,y) \in R \implies y \in C^{\mathcal{I}}\}$

A Description Logic Knowledge base consists of a set of axioms about Instances, concepts (potentially defined in terms of complex concept expressions and relations). The first type of axioms can be used to describe instances. In particular, axioms can be used to state that an instance belongs to a concept, that two instances are in a certain relation. It is easy to see, that these axioms can be used to capture case descriptions as labeled graphs. The other type of axioms describe relations between concepts and instances. It can be stated that one concept is a subconcept of the other (all its instances are also instances of this other concept). Further, we can define a relation to be a subrelation or the inverse of another relation. These Axioms are used to formalize the legal ontology described in the last section. The formal definition of axioms can be found in the table below.

DL Axiom	Semantics
$C(x)$	$x^{\mathcal{I}} \in C^{\mathcal{I}}$
$P(x,y)$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in P^{\mathcal{I}}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$P \sqsubseteq R$	$P^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
$P \equiv R^{-}$	$P^{\mathcal{I}} = \{(x,y) \mid (y,x) \in R^{\mathcal{I}}\}$

The formal semantics of concepts and relations as defined by the interpretation into Δ can be used to automatically infer new axioms from existing definitions. In particular, given an ontology and a number of instance related axioms, we can automatically determine whether an instance belongs to a certain concept based on the expression defining the concept.

4.2. Determining Potential Liability

The idea of the automatic determination of potential liability is now to describe concepts related to liability in terms of necessary and sufficient conditions for liability using the logic introduced above. These conditions are phrased in terms of relations to other actors in a case. An example for such a definition is the following:

$$LiableForProperty \equiv \exists owns.(\exists causes.(damage \sqcap \exists affects.plaintiff))$$

It says that a person can in principle be held liable for property according to Dutch law if the person owns something that caused a damage to the accusing party. Using additional background knowledge from the different ontologies, this abstract definition can be matched with the case description shown in figure 3. The riding school owns the horse which caused an injury. Further, the accusing party suffered the injury. Using the additional information that an injury is a special case of a damage ($injury \sqsubseteq damage$) and that suffers is the inverse relation to affects ($suffers \equiv affects^{-}$), we can see that the riding-schoolfulfills all the necessary conditions of being potentially liable for property.

A characteristic of reasoning about potential liability is that often there are different parties that could be held liable and even the same party can often be held liable on different grounds. In the following, we show how these features are implemented in our approach using the concept of liability for a subordinate. This concept is defined in the following way:

$$LiableForSubordinate \equiv \exists hasSubordinate.PersonallyLiable$$

The definition says that someone can be held liable for a subordinate if there is a hasSubordinate relation to someone who is potentially personally liable in the case. Using the additional information that being employed is a special case of being a subordinate ($employs \sqsubseteq hasSubordinate$) we know that the riding school would also fulfill this requirement provided that the teacher, who is employed at the school is potentially personally liable in the case. This is checked by the system on the basis of the corresponding

definition of the concept of being personally liable. This concept is a quite generic one and covers a number of cases, only two of which we show here:

$$\begin{aligned} PersonallyLiable \equiv\ & legal - person \sqcap \exists causes.(damage \sqcap \exists affects.plaintiff) \\ & \sqcup \exists handles.(object \sqcap \exists causes.damage \sqcap \exists affects.plaintiff) \\ & \sqcup ... \end{aligned}$$

The definition says that someone is potentially personally liable if either he is a person in the legal sense and he has caused some damage that affected the accusing party or handles an object (in the legal sense) that has caused some damage to the accusing party, and so on. These two cases are interesting, because they require to explicitly make a distinction between legal persons and legal objects. If this distinction would not be made, the horse in our example would be classified as being personally liable. The definition above correctly determines that the teacher in our example is potentially personally liable according to the second case mentioned in the definition. Further, using the additional information that objects are not legal persons ($object \sqsubseteq \neg legal - person$) the system concludes that the horse is not liable for anything.

4.3. Our Axiomatization of Dutch Tort Law

We performed a complete axiomatization of different forms of liability in Dutch tort law that allows us to automatically determine liability in complex case definitions. We tested the reasoning on a number of real world cases of which the one described above is the most simple using the FaCT reasoning system [8] in combination with the OilEd Ontology editor [3]. Some cases require very complex reasoning including relation hierarchies, inverse relations and global constraints and can take up to a couple of minutes. We are planning to move to a different reasoning system to improve the performance.

We tested the reasoning capabilities of the ontology on a number of example cases taken from legal text books. More details about these cases can be found in [11] and on the BEST web site (http://www.best-project.nl/cases/). The definitions of different forms of liability in the ontology was rich enough to cover almost all aspects of these example cases. We will discuss some of the current limitations in the next section.

5. Conclusions

Our claim is that logical reasoning can be used to mediate between the terminology used by laymen to describe legal cases and the terminology used in legal documents. The results presented in this paper provide a first important step in this direction. By being able to automatically detect relevant law articles based on a structured case description, we have made a first step towards determining relevant legal documents. The second step required is to link legal documents to combinations of law articles either by analyzing direct references to articles or through the occurrence of terms characteristic for a certain article.

While we have been able to build a computational ontology of a Dutch tort law that can determine relevant articles, the conclusions drawn based on this ontology are often impartial in the sense that certain relevant aspects of the case, such as knowledge

of dangers, attempts to minimize the danger, whether an object is faulty or not have been abstracted away in order to make the approach feasible. In order to really come up with a judgment on the case, these aspects have to be further investigated by a legal professional. We nevertheless belief that the judgement that can be done on the basis of the ontology is useful for the layman to better understand the options available. On the other hand, we avoid the danger of producing results that are pure speculation. A real evaluation of the approach will only be possible, however, when the second step – the retrieval of documents – is implemented.

References

[1] M. C. Aaron. Finding settlement with numbers maps, and trees. In M. L. Moffit and R. C. Bordone, editors, *The Handbook of Dispute Resolution*. Jossey-Bass, 2005.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] S. Bechofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: A reason-able ontology editor for the semantic web. In F. Baader, G. Brewka, and T. Eiter, editors, *KI 2001: Advances in Artificial Intelligence*, pages 396–408. Springer, 2001.

[4] G. Betlem. *Civil Liability for Transfrontier Pollution*. Graham and Trotman, 1993.

[5] P. Casanovas, N. Casellas, J.-J. V. M. Poblet, F. Ramos, J. Gorroogoitia, J. Contreras, M. Blï£¡quez, and R. Benjamins. Iuriservice ii: Ontology development and architectural design. In *Proc. of the Tenth International Conference on Artificial Intelligence and Law*, pages 188–194, Bologna, Italy, 2005.

[6] R. Fisher and W. Ury. *Getting to YES: Negotiating Agreement Without Giving In*. Haughton Mifflin, 1981.

[7] A. Gangemi, D. Pisanelli, and G. Steve. An ontological framework to represent norm dynamics. In R. Winkels, T. van Engers, and T. Bench-Capon, editors, *Proceedings of the Second International Workshop on Legal Ontologies*, 2001.

[8] I. Horrocks. Fact and ifact. In *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 133–135, 1999.

[9] A. Lodder and J. Zeleznikow. Developing an online dispute resolution environment: Dialogue tools and negotiation support systems in a three-step model. *Harvard Negotiation Law Review*, 10:287–338, 2005.

[10] H. Stuckenschmidt, A. de Waard, R. Bhogal, C. Fluit, A. Kampman, J. van Buel, E. van Mulligen, J. Broekstra, I. Crowlesmith, F. van Harmelen, and T. Scerri. Exploring large document repositories with rdf technology - the DOPE project. *IEEE Intelligent Systems*, 2004.

[11] R. van Laarschot. Ontology-based knowledge modelling in dutch civil law. Master's thesis, Vrije Universiteit Amsterdam, August 2005.

[12] W. van Steenbergen. Doorverwijzing naar mediation vanuit de rechtspraak: Contradictio in terminis of een goede manier om de rechterlijke macht te ontlasten? *Nederlandsch Juristenblad*, 2005.

[13] J. Zeleznikow and A. R. Lodder, editors. *Fundamentals of Providing Negotiation Support Online: The Need for Developing BATNAs*. Wolf Legal Publishers, 2005.

This page intentionally left blank

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Validating an Automated Evaluation Procedure for Ontology Triples in the Privacy Domain

Peter Spyns [a,1] and Giles Hogben [b]
[a] *Vrije Universiteit Brussel - STAR Lab, Belgium*
[b] *European Commission Joint Research Centre - IPSC, Ispra, Italy*

Abstract. In this paper we validate a simple method to objectively assess the results of extracting material (c.q. triples) from text corpora to build ontologies. The EU Privacy Directive has been used as corpus. Two domain experts have manually validated the results. Several experimental settings have been tried. As the evaluation scores are rather modest (sensitivity or recall: 0.5, specificity: 0.539 and precision: 0.21), we see them as a baseline reference for future experiments. Nevertheless, the human experts appreciate the automated evaluation procedure as sufficiently effective and time-saving for usage in real-life ontology modelling situations.

Keywords. Ontology Evaluation, Privacy Ontology Mining

1. Introduction and Background

The development of the Semantic Web (of which ontologies constitute a basic building block) has become a very important research topic for the (future) knowledge based society. However, the process of conceptualising an application domain and its formalisation require substantial human efforts. Therefore, techniques applied in human language technology (HLT) and information extraction (IE) are used to create or grow ontologies. Work is still in progress - recent overviews of the state of the art (in particular for machine learning techniques) can be found in [1,2]. Even in the ideal case that (semi-)automated HLT and IE methods have become mature, there still remains the problem of assessing and evaluating the results. Recent proposals for evaluation methods are [1,3].

All these approaches share basically the same problem, i.e. how to determine a gold standard valid for various application situations. Rare are the experts who are willing to devote their time to validate output generated by a machine or establish in agreement a gold standard for a particular technical domain. Therefore, we want to define a light-weight assessment procedure that is easy to understand and apply by "‘standard knowledge workers"’ (basically a domain expert, a computer scientist, an engineer, ...) outside the specialised academic environment.

[1] correspondence to: Peter Spyns, Vrije Universiteit Brussel - STAR Lab, Pleinlaan 2, Gebouw G-10, B-1050 Brussel, Belgium. Tel.: +32-2-629 3753; Fax: +32-2-629 3819; Email: Peter.Spyns@vub.ac.be.

The method should be generally applicable (any kind of text miner, any kind of text) and able to provide a rough but good enough and reliable indication whether or not results of a text miner on a particular corpus are worthwhile. Typical of our approach will be that only the corpus (lemmatised [1] but otherwise unmodified) constitutes the reference point, and not an annotated corpus or another gold standard ontology. The reason being that these artifacts require quite some human effort to be built. However, the evaluation procedure itself should first be validated, and therefore we have to rely on human experts. This paper focusses on this latter aspect.

As a test corpus, we used the European Data Protection Directive (95/46/EC) in the application domain of the PRIME project [2]. The use-case for the final ontology is the creation and processing of machine-readable data-handling practice statements (privacy policies) by data processors within the European Union jurisdiction. The semantics of the ontology will support the evaluation of practice statements when deciding whether to release data, either by users or automated evaluators mandated by users (user-agents). It will also be used to support administators in creating machine-readable policies, which are legally compliant.

The remainder of this paper is organised as follows. The next two sections present the material (section 2) and methods (section 3): the lexicometric scores are shortly explained in section 3.1, after which the procedure to evaluate triples is discussed (section 3.2) as well as its validation method (section 3.3). The outcomes of the lexicometric measures (section 4.1) as well as the triple scoring procedure (section 4.2) are described and discussed subsequently in section 5. Related work is outlined in section 6. Indications for future research are given in section 7, and some final remarks (section 8) conclude this paper.

2. Material

The *memory-based shallow parser for English*, being developed at CNTS Antwerp and ILK Tilburg [4], has been used. Semantic relations that match predefined syntactic patterns have been extracted from the shallow parser output. Additional statistics using normalised frequencies and probabilities of occurrence are calculated to separate noise (i.e. false combinations generated by chance during clustering) from genuine results. More details on the linguistic processing can be found in [5].

The *Privacy corpus* (a single long document) consists of 72,1K words. It constitutes the EU directive on Privacy (95/46/EC of 18 dec 2000 - English Version) that has to be adopted and transformed into local legislation by every Member State. This was chosen because the document provides the legal context in which the application domain's use cases will operate. The ontology is used to model privacy enhanced access control policies (access rules controlling data processing events) within Europe. E.g., it will be able to provide an automated judgement as to the legality of a data processing event within the EU.

The CNTS text miner has been applied to this corpus. After some format transformation, it outputs "subject-verb-object" triples, such as <*third_country, ensure,*

[1] Lemmatise means to reduce words to their base form. E.g., working, works, worked → work. Incidentally note that in this paper, the terms 'word', 'term', and 'lemma' are used interchangeably.

[2] See http://www.prime-project.eu.org

level_of_protection>, "noun phrase-preposition-noun phrase" triples such as <*Treaty, on, European_Union*> and subject-verb-prepositional object triples, such as <*controller, establish, in_Member_State*>, in total 1116 triples. In addition, the *Wall Street Journal (WSJ) corpus* (a collection - 1290K words - of English newspaper articles) serves as a "neutral" corpus that is to be contrasted with the specific technical vocabulary of the Privacy Directive.

An off-the-shelf available *lexicographic program* (WordSmith v4) has been used to create the frequency lists. Further manipulation of exported WordSmith files and calculations are done by means of small scripts implemented in *Tawk* v.5 [6], a commercial version of (G)awk, in combination with some manipulations of the data in *MS Excel*.

3. Methods

An ontology is supposed to represent the most relevant concepts and relationships of a domain of discourse. The terms lexicalising these concepts and relationships are to be retrieved from a corpus of texts about the domain. The key question is how to determine in an automated way which are the important terms (section 3.1). In addition, an algorithm is needed to distinguish relevant combinations (i.e. two concepts in a valid relationship - a triple) from irrelevant ones (section 3.2). Both steps have to be validated. In this paper, we'll primarily discuss the validation of triples (section 3.3), the validation of the relevant terms having been presented elsewhere [7].

3.1. Determining the Relevant Words

The central notion linking everything together is "frequency class" (FC), i.e. the set of (different) lemmatised words that appear n times in a document d. E.g., for the Privacy Directive, there are 416 words that appear only once (FC 1), and there is one word that appears 1163 times (FC 1163). According to Zipf's law [8], the latter one ('the') is void of meaning, while the former ones (e.g., 'assurance') are very meaningful, but may be of only marginal interest to the domain. Subsequently Luhn [9] introduced the notion of "resolving power of significant words", by defining intuitively a frequency class upper and lower bound. The most significant words are found in the middle of the area of the FCs between these boundaries.

We propose to calculate whether the FC is relevant or not. Only if a FC is composed by 60%[3] or more of relevant words, the FC is considered to be relevant. A word is said to be relevant or not based on the outcome of a statistical formula that compares two relative proportions. Calculations have been done for 95% and 99% confidence level. The relevant words are used as the gold standard for the precision and recall metrics, which has otherwise to be created by human experts. The coverage (involving all FCs) and accuracy (involving only relevant FCs) metrics indicate to which degree the vocabulary of a triple mining tool coincides with the set of relevant words. Due to space restrictions, we have to refer the reader to [7] for more details on the metrics.

[3]Currently, this threshold has been choosen arbitrarily.

3.2. Determining the Relevant Triples

Again, the relevant terms are used as reference (one set computed with a 95% confi dence level and one with a 99% level). Several basic scenarios have been set up in which a triple (= Subject-Predicate-Object) is considered relevant (see also Table 3 and Table 4):

- Two (A1, B1) or three (C1,D1) of the three triple parts contains a term statistically relevant.
- The degree with which a triple lexically overlaps with terms of the reference set depasses a certain level: 65% (X1, Y1), 70% (X2, Y2), and 90% (X3, Y3).
- The degree of lexical overlap is combined with the number of triple parts containing a relevant term.

We did not use a stopword list, as this list might change with the nature of the corpus, and as a preposition can be potentially relevant (unlike e.g. in IE applications) since they are included in the triples automatically generated.

3.3. Validating the (Relevant) Triples

Two experts have been asked to independently validate the list of triples as produced by the text miner. One has been a privacy data commissioner and still is a lawyer while the other an knowledge engineer specialised in the fi eld of privacy and trust. They have marked the list of triples with '+' or '-' indicating whether or not the triple is valid, i.e. usefull in the context of the creation of a privacy ontology. It allows us to evaluate the "goodness of fi t" of the automated procedure compared to a human reference. Of course, it would have been better if more than two human experts would have been involved, but many experts are reluctant to perform this kind of validation as it is quite tedious and boring.

4. Results

In total 1116 triples have been generated by the CNTS unsupervised text miner: 278 have a verb, 557 a preposition and 286 a verb-preposition combination on the predicate position. 49 FCs are considered as relevant. It is important to remember that the scores mentioned in section 4.1 measure the intersection of corpus vocabulary considered relevant with vocabulary derived from a series of triples generated automatically. The scores given in section 4.2 concern the goodness of fi t of an automatic evaluation procedure compared to (two) human experts who have evaluated the same set of triples.

4.1. Word Relevance

When only taking into consideration the 49 relevant FCs of the privacy corpus, the recall is 95,58%, the precision is 96,29% and the accuracy is 95,04%. The κ value comparing the miner results with the statistical results (again only for the relevant FCs) is 0,69%, which signifi es a good agreement. These good results are probably obtained because the FCs from 1 till 10 are not considered relevant. However, they contain the bulk of the vocabulary. From another experiment (using the EU VAT Directive as corpus) in which

an expert manually had created a list of relevant terms, there is a strong indication that the lower FCs cannot be ruled out [7]. An option would be to relax the 60% treshold. When taking all the privacy FCs into consideration (which is a too simplistic relaxation), the recall is 89,91%, the precision 27,43% and the coverage 79,88%. The κ value now is 0,14% (very close to contradiction).

4.2. Triple Relevance

The 1116 triples have been rated by two human experts ('+' for appropriate vs. '-' for non appropriate). Their ratings resulted in a κ value of -0,0742 (= almost contradiction - see Table 1), which means that they agree in a way even less than expected by chance. For subsequent tests, only the triples commonly agreed upon (in a positive (112) and negative (463) sense) have been retained as the reference.

Table 1. Inter experts agreement: $\kappa = -0,0742$.

	Expert 2 +	Expert 2 -	
Expert 1 +	112	292	404
Expert 1 -	249	463	712
	361	756	1116

Some examples of triples are shown in Table 2. E1 and E2 correspond to how expert 1 resp. expert 2 rated a triple. X% and Y% indicate the absolute lexical overlap percentage (see below). In addition, various different evaluation scenarios have been investigated - see Table 3 for the exact settings.

Table 2. Some triples and their status in eight evaluation scenarios - see Table 4.

triple	E1	E2	X%	Y%	A1	B1	C1	D1
breach, of, right	+	+	66	66	+	+	+	-
functioning, of, internal_market	-	+	53	33	+	+	+	+
recording, of, personal_data	+	+	66	66	+	+	+	+
rise, to, damage	+	+	33	33	-	-	-	-
chairman_term, be, two_year	+	-	0	0	-	-	-	-
exercise, be, public_administration	+	-	44	44	+	-	+	-
own_initiative, make, recommendation_on_matter	+	+	9	0	+	-	-	-
course_activity, fall, outside_scope	+	+	48	35	+	-	+	-
staff_supervisory_authority, even, after_employment	-	-	28	28	-	-	-	-

In a first stage, confidence levels (i.e. the confidence level, with which the separate lemmas are considered to be relevant) of 95% and 99% have been combined with the fact whether two ('-++', '+-+' and '++-') or three ('+++') constituting elements of a triple contain relevant vocabulary (see Table 3). E.g., in the scenario A1 (two triple parts have to contain a lemma relevant with a confidence of 95%) 1012 triples are accepted as appropriate ('+'), while 104 are rejected ('-').

Subsequently, the qualitative criterium (does the triple contain a relevant word) has been replaced by a score indicating how many characters of the three triple parts (ex-

Table 3. Some of the evaluation scenarios and the corresponding results.

experimental settings	label	-	+	+++	++-	+-+	-++
3/3 (expert 1)	E1	712	404				
3/3 (expert 2)	E2	755	361				
95% conf, 2/3	A1	104	1012	637	59	224	92
95% conf, 65% score	X1	346	770				
95% conf, 2/3, 65% score	A2	104	770	594	34	77	65
95% conf, 3/3	B1	479	637	637			
95% conf, 70 %score	X2	582	534				
95% conf, 3/3, 70% score	B3	479	534	534			
99% conf, 2/3	C1	148	968	497	73	267	131
99% conf, 65% score	Y1	464	652				
99% conf, 2/3, 65% score	A2	148	652	445	42	79	86
99% conf, 3/3	D1	619	497	497			
99% conf, 70% score	Y2	720	396				
99% conf, 3/3, 70% score	D3	619	396	396			

pressed as an averaged percentage) are matched by words statistically relevant. E.g., the triple $< rule, establish, by\_national\_competent\_body >$ receives a score of 89 as only 'competent' is not a relevant word with a 95% confi dence level (89 = ((4/4)*100 + (11/11)*100 + (17/25)*100)/3) [4]. Finally, 22 combinations have been tried - see Table 4. E.g., if there is no perfect match for the entire triple (that would equal a 100% score), one could still establish a treshold (e.g., 90%) that on the one hand relaxes the '+++' criterium and on the other hand still rejects matches that are too partial (scenarios B4 and D4). Note in that respect that 66% corresponds to a perfect match of two of the three triple elements.

Table 4. Sensitivity (Se), precision (P), and specificity (Sp) values.

setting	label	Se	P	Sp
95% cf, 2/3	A1	0,955	0,211	0,133
95% cf, 3/3	B1	0,580	0,205	0,455
99% cf, 2/3	C1	0,910	0,212	0,179
99% cf, 3/3	D1	0,464	0,211	0,578
95% cf, 65%	X1	0,723	0,208	0,332
95% cf, 70%	X2	0,50	0,208	0,539
95% cf, 90%	X3	0,383	0,117	0,645
99% cf, 65%	Y1	0,616	0,205	0,421
99% cf, 70%	Y2	0,410	0,212	0,630
99% cf, 90%	Y3	0,321	0,226	0,734

setting	label	Se	P	Sp
95% cf, 2/3, 65%	A2	0,723	0,168	0,136
95% cf, 2/3, 70%	A3	0,508	0,157	0,341
95% cf, 2/3, 90%	A4	0,392	0,226	0,673
99% cf, 2/3, 65%	C2	0,723	0,176	0,179
99% cf, 2/3, 70%	C3	0,410	0,236	0,678
99% cf, 2/3, 90%	C4	0,321	0,263	0,781
95% cf, 3/3, 65%	B2	0,553	0,163	0,311
95% cf, 3/3, 70%	B3	0,508	0,178	0,431
95% cf, 3/3, 90%	B4	0,392	0,164	0,431
99% cf, 3/3, 65%	D2	0,428	0,151	0,416
99% cf, 3/3, 70%	D3	0,410	0,175	0,531
99% cf, 3/3, 90%	D4	0,321	0,142	0,531

We have also calculated the sensitivity or recall (indicates the power of the evaluation procedure to accept true positive facts), specifi city (indicates the power of the evaluation

[4] A slight imprecision occurs due to the underscores that are not always accounted for.

procedure to reject true negative facts) and precision (indicates to which extent the results obtained are correct) for the triples rated in the same way by both experts - see Table 4.

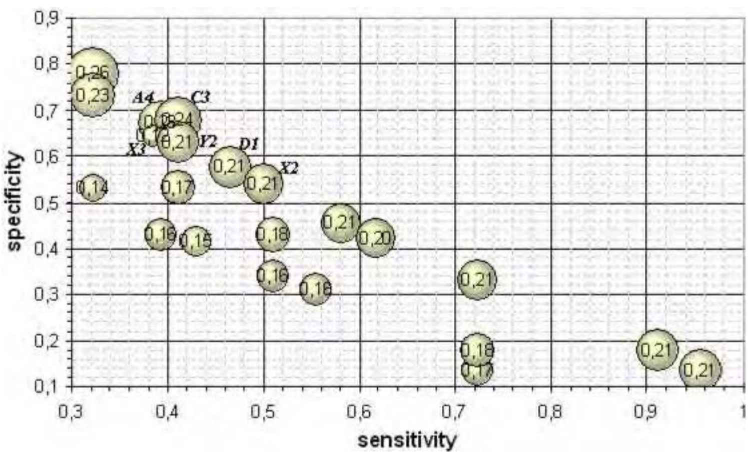

Figure 1. Sensitivity versus specificity and precision.

Figure 1 graphically represents the data of Table 4. The size of the bubbles (each representing a different experimental setting - e.g. X3, A4, C3, Y2, D1 and X2) indicate the precision (value inside the bubble) for a combination of sensitivity (X-axis value) and specifi city (Y-axis value) scores.

5. Discussion

It is to be expected that taking care of synonyms will improve the scores, but it is unclear to what extent. On the other hand, synonym lists or domain specifi c vocabularies with explanatory glosses will not always be available. In those cases, a corpus will be the only source of information (next to human experts). Both the human experts jointly examined their initial evaluation. They stated that many disagreements were due to a difference in understanding of the basis criteria. This could be improved by a clearer statement of evaluation criteria in combination with a test run on a small number of terms where evaluators explain each choice and are given the opportunity to comment on any differences (although to each other to avoid bias). E.g., one evaluator erroneously removed all object terms containing a preposition because he decided they were not strictly nouns, whereas the other did not. This difference in criteria would quickly be detected and removed by the test run and subsequent check. Unlike for classical IE evaluation settings, inter-rater disagreement is less problematic as it reflects a different position regarding the ontology depending on the (type of) stakeholder involved. For the moment, hierarchical and other relationships are not yet assessed. We have to bare in mind that the entire automatic triple evaluation procedure is "blind", i.e. mainly based on term frequencies and string overlap instead of domain knowledge.

5.1. Word Relevance

In a previous experiment [7], we've tested our metrics on the EU VAT Directive. When comparing the machine reference with a human reference (a list of terms produced by human experts), a κ value of 0,758 has been reached, indicating a good agreement. Therefore, we assumed that, in case of the Privacy Directive, a similar result, i.e. the metrics approximate human experts, would be obtained (as we don't dispose yet of a list of privacy terms made by human experts).

5.2. Triple Relevance

Minimally the sensitivity and specifi city values should be higher than 0,5. This happens only with scenario X2 (95% confi dence and 70% score levels). However, the precision only reaches 0,208. We see that the scoring algorithm mainly enhances the sensitivity. The confi dence level has a positive impact on the specifi city and a negative one on the sensitivity. In the situation of ontology engineering we estimate that a high specifi city is more interesting than a high sensitivity (less false positives at the detriment of less true positives): a relevant triple might be missed in order to have less rubish triples. The rationale is that it is probably more effi cient to reduce the extent of the material ontology engineers have to check and reject compared to their effort needed to detect missing material. Therefore, scenario C3 reaching a precision of 0,236 (with of sensitivity and specifi city of 0,41 and 0,678 resp.) can be an alternative. We consider X2 and C3 as scenarios that provide *baseline* results as the precision scores are too low.

The experts stated that too many irrelevant results are produced - the text miner not being able to skip over sections that are only of marginal interest for the privacy topic. To remedy this, they have suggested the following improvements:

- The background ("neutral") corpus (here the WSJ) is key in defi ning what constitutes relevance. However, legal documents have many terms which are relevant to the legal domain in general, but <u>not</u> relevant to the particular legal domain under consideration. In future experiments using legal documents it is recommended to use a background corpus of terms taken from a set of European legal documents. This would eliminate a large number of classifi cation discrepancies between experts themselves, and between the experts and the unsupervised miner. For example, the term "Member State" is highly relevant to European Legislation in general, but has no specifi c relevance to the privacy domain. This is an example of a term that was judged higly relevant by the miner, but totally irrelevant by the experts.
- European legislative texts have a uniform structure and therefore lend themselves to a pre-processing stage where irrelevant material can be quickly and repeatably removed. For example, in creating a privacy ontology for user-agents, sections about member states' obligations to inform the European Commission when legislation has been implemented are clearly not relevant. It is suggested that the text be pre-processed according to a well-defi ned set of criteria (e.g., because our ontology is applied to the modelling of normative rules, we were able to remove the "whereas sentences" in the preprocessing step, using them only for later grounding of term semantics) before being processed by the text miner. This will significantly reduce the number of irrelevant triples.

- An issue not addressed by the above process is that of abstraction. Human experts extracting terms from a corpus are able to amalgamate synonyms and instances of higher level concepts where the use of lower level terms is of no use to the application domain. For instance, the privacy directive gives a list of data types which it is prohibited to collect without the data subject's consent. To a human expert, these classes of data are clearly what is known as "sensitive data". The inclusion of a synomym dictionary would go some way towards term abstraction although it can only take account of equivalence and not subclass relationships between terms.

6. Related Work

Reports on previous experiments contain additional details on the unsupervised miner [5]. The method and previous quantitative experiments have been presented in [7]. To the best of our knowledge, so far only one other approach addresses the quantitative and automated evaluation of an ontology by referring to its source corpus [10]. Others have evaluated methods and metrics to select the most appropriate terms (e.g, [11]) for building an ontology. However, they don't evaluate entire triples. Various researchers are working on methods to evaluate results of ontology learning and population [1,3].

Brewster and colleagues have presented a probabilistic measure to evaluate the best fi t between a corpus and a set of ontologies as a maximised conditional probability of fi nding the corpus given an ontology [10]. Unfortunately, no concrete results or test case are presented. Next to that, there is the work of *Sabou* who, in her latest work, tries to learn ontologies for web services from their descriptions. Although the practical aspects of her work on the ontology learning aspects are quite tailored towards the application domain, the evaluation method resembles well ours. She has "established a one-to-one correspondence between phrases in the corpus and derived concepts [12], so that our lexicometric score are comparable to her ontology ratios.

7. Future Work

The core topic concerns research on the goodness of fi t of the evaluation procedure as an approximation of the human expert behaviour. This necessitates the involvement of minimally two human experts to account for inter-rater (dis)agreement. Some axes for future work emanate from the discussion section:

- Study alternative term selection methods available from the domains of lexico- and terminography, quantitative linguistics or library information science - e.g., the TF/IDF and domain relevance and consensus metrics. [5] - as well as alternative heuristics to asses entire triples.
- Study the "move" towards the conceptual level, which necessitates the integration of semantic distance measures such as the WordNet similarity functions [13]. Brewster et al. add two levels of WordNet hypernyms [10, p.166] for that purpose. That implies that (novel) compound terms should be assigned a semantic interpretation as is done e.g., by Navigli and Velardi [11].

---

[5] Up till now, it was pointless to use these as the corpus consists of one (resaonably large) document.

8. Conclusion

We have presented a validation effort for an easily to apply automatic evaluation procedure for triples as material for a privacy ontology to be created. Compared to a human reference, the automatic evaluation procedure is able in more or less half of the cases to successfully accept an appropriate triple and reject an irrelevant triple. The precision score however is too low. Nevertheless, the automatic evaluation procedure is considered practically useful by the human experts to speed up the ontology creation process. The current outcomes can be considered as a baseline reference for further experiments.

Acknowledgements

This research has been fi nanced by the Flemish OntoBasis project (IWT GBOU 2001 #10069) and the EU FP6 IP PRIME (IST 2002-507591). It represents the view of the authors only. We are particularly indebted to dr. Marie-Laure Reinberger (Universiteit Antwerpen - CNTS) who has produced the pricavy triples and to drs. John Borking (Borking Consultancy, The Netherlands), who was the second domain expert.

References

[1] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, editors. *Ontology Learning from Text: Methods, Applications and Evaluation*. IOS Press, Amsterdam, 2005.

[2] M. Shamsfard and A. Barforoush. The state of the art in ontology learning: a framework for comparison. *Knowledge Engineering Review*, 18(4):293 – 316, 2003.

[3] J. Hartmann, P. Spyns, D. Maynard, R. Cuel, M. Carmen Suarez de Figueroa, and Y. Sure. Methods for ontology evaluation. KnowledgeWeb Deliverable #D1.2.3, 2005.

[4] Sabine Buchholz, Jorn Veenstra, and Walter Daelemans. Cascaded grammatical relation assignment. In *Proceedings of EMNLP/VLC-99*. PrintPartners Ipskamp, 1999.

[5] Marie-Laure Reinberger and Peter Spyns. *Ontology Learning from Text: Methods, Applications and Evaluation*, chapter Unsupervised Text Mining for the Learning of DOGMA-inspired Ontologies. IOS Press, Amsterdam, 2005.

[6] Thompson Automation Software, Jefferson OR, US. *Tawk Compiler*, v.5 edition.

[7] P. Spyns and M.-L. Reinberger. Lexically evaluating ontology triples automatically generated from text. In A. Gómez-Pérez and J. Euzenat, editors, *Proceedings of the second European Semantic Web Conference*, volume 3532 of *LNCS*, pages 563 – 577. Springer Verlag, 2005.

[8] George K. Zipf. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley, 1949.

[9] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159 – 195, 1958.

[10] Christopher Brewster, Harith Alani, Srinandan Dasmahapatra, and Yorick Wilks. Data driven ontology evaluation. In N. Shadbolt and K. O'Hara, editors, *Advanced Knowledge Technologies: selected papers*, pages 164 – 168. AKT, 2004.

[11] Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151 – 179, 2004.

[12] Marta Sabou, Chris Wroe, Carole Goble, and Gilad Mishne. Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *Proceedings of the 14th International World Wide Web Conference*, 2005.

[13] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. In *The Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, 2004.

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

An Ontological Approach for the Management of Rights Data Dictionaries

Roberto García and Jaime Delgado
Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{roberto.garcia, jaime.delgado}@upf.edu

Abstract. RDDOnto provides an ontological approach to the Rights Data Dictionary (RDD) part of MPEG-21, one of the main Intellectual Property Rights (IPR) Management standardisation efforts. In order to build the ontology, the terms defined in the RDD specification have been modelled using OWL, trying to capture the greatest part of its semantics. The ontology allows formalising a great part of the standard and simplifying its verification, consistency checking and implementation. During the RDDOnto construction, some integrity problems were detected, which even have led to standard corrigenda. Additional checks were possible using Description Logic reasoning in order to test the standard consistency. Moreover, RDDOnto is now helping on how new terms can be added to the RDD and to integrate the RDD with other parts of MPEG-21 also mapped to OWL. Finally, there are the implementation facilities provided by the ontology. They have been used to develop MPEG-21 licenses searching, validation and checking. Existing ontology-enabled tools as semantic query engines or logic reasoners facilitate this.

Keywords. Digital Rights Management, Intellectual Property, Ontology, Semantic Web

1. Introduction

The number of online marketplaces has grown in recent years and will continue to expand in the future. Content companies consider unauthorised use or reproduction of digital content a serious problem. The goal of a Digital Rights Management (DRM) system is to enforce licenses between a content provider and a consumer that define rules about authorised use of managed content.

The Moving Pictures Expert Group (MPEG) [1] is the ISO/IEC working group in charge of developing standards for the coded representation of digital audio and video. Among other standards, the group is working on the MPEG-21 standard [2] with the objective of developing a standardised multimedia framework. The fifth part of MPEG-21 specifies a Rights Expression Language (REL) [3] and the sixth one an associated Rights Data Dictionary (RDD) [4].

The rights language introduces ways to assign rights expression to digital goods or services and to control usage and access. The two constitutive factors in a rights language are its syntax and semantics. The term syntax refers to the grammar rules, which apply to the language's vocabulary, whereas the term semantics refers to the

meaning of valid sentences in the language. Each rights expression language includes a rights vocabulary or dictionary, which defines the permitted words and their semantics.

There are other initiatives for rights expression languages, but there is just one that defines also a rights data dictionary. It is the ODRL initiative [5]. However, the ODRL Data Dictionary, on the contrary to MPEG-21 RDD, was not developed as an ontology that provides the semantics of the REL terms. ODRL DD is a XML Schema, like MPEG-21 REL and ODRL REL, and it extends the generic elements defined in the ODRL REL but from a purely syntactic point of view.

Therefore, we have centred our research on applying an ontological approach to rights data dictionaries on MPEG-21 RDD. It is generic so it can be applied to other dictionaries like the one in MPEG-21. For dictionaries like the one in ODRL we have explored other methodologies that also try to move them to an ontology space. The common part is that the final objective in all cases is to use web ontologies as an integration framework where all these initiatives and approaches can be connected, made interoperable and enriched from the new facilities provided by ontologies.

2. The Rights Data Dictionary Ontology

The objective of RDDOnto is to translate the RDD terms descriptions from its current textual representation in the dictionary to a machine processable representation using the semantic web paradigm.

The set of all predefined classes and properties are the building blocks provided by the OWL [6] and RDF/S (RDF plus RDFSchema) [7,8] frameworks. These building blocks are used to construct Semantic Web ontologies, i.e. sets of restrictions to the basic RDF elements. These restrictions can be automatically validated in order to test that a particular RDF description conforms to the semantics of the particular domain captured by the ontology.

In the next subsections, we will analyse RDD and then detail how first RDF/S and afterwards OWL frameworks can be used to capture RDD terms definitions and a great part of their semantics. RDF/S is capable of modelling only a fraction of the RDD semantics. This fraction is augmented when the constructs introduced by OWL are also used. Therefore, two versions of the ontology can be produced. The simpler one uses RDF/S and the more complex one uses OWL.

2.1. RDD Specification analysis

The RDD Specification [4] defines a set of terms, the "words" in the vocabulary. The RDD Specification is self contained so all the terms that it uses, even the relating terms, are defined in it. For each term, its description is composed by a set of attributes:

- **Headword**: the term name. It must appear in the term description.
- **Synonym**: some alternative names. It is not mandatory.
- **Definition**: a short text that defines the term.
- **MeaningType**: allowed values are: Original, PartlyDerived and Derived.
- **Comments**: extended textual information about the term. It is not mandatory.
- **Relationships**: this attribute lists the relationships, from a set of predefined ones, among this term an other terms. They are used to specify the term semantics from different points of views. The relations are classified in the following categories:

- **Genealogy**: these relations give a semantic point of view similar to that from Semantic Networks [9], i.e. inheritance, relations domain and range, etc. The relations are: *IsTypeOf*, *IsA*, *Is*, *IsEquivalentTo*, *IsOpposedTo*, *IsPartOf*, *IsAllowedValueOf*, *HasDomain*, *HasRange* and *IsReciprocalOf*.
- **Types**: they are enumerated using *HasType* and its reciprocal *IsTypeOf*.
- **Membership of Sets**: the relating term from members to sets, *IsMemberOf*.
- **Family** these relationships connect an *ActType* and the terms that it begets through the application of the Context Model semantics. E.g. *BegetsAgentType*.
- **ContextView**: the group of relationships describing the attributes of a specific *ContextType* using the Context Model semantics.

2.2. RDD to Web Ontology Mappings

From the RDD Specification analysis two kinds of attributes can be detected. The first group is composed by those attributes with unstructured values, i.e. textual values. They can be easily mapped to predefined or new RDF properties with textual (literal) values.

The first option is to try to find predefined RDF properties that have the same meaning that the RDD term attributes that are being mapped. When this is not possible, the RDFS constructs will be used to define new RDF properties to which the corresponding attributes will be mapped. These properties are defined in the RDDOnto namespace, "rddo".

The mappings of this kind are shown in Table 1. Note that the Dublin Core [10] RDF Schema is also reused in RDDOnto. The Dublin Core (DC) metadata element set is a standard for cross-domain information resource description. The DC RDF Schema implements the Dublin Core standard.

Table 1. Mappings for the RDD attributes with text value

RDD Attribute	RDF Property	Kind of RDF property
Headword	rdf:ID	Predefined in RDF
Synonym	rddo:synonym	New property defined in RDDOnto
Definition	dc:description	Predefined in Dublin Core RDFS
MeaningType	rddo:meaningType	New property defined in RDDOnto
Comments	rdfs:comment	Predefined in RDFS Schema

The other kind of attribute is the *Relationships* one. Its value is not textual. Firstly, it is categorised into five groups: *Genealogy*, *Family*, *ContextView*, *Types* and *Membership of Sets*. Each of these groups is composed by a set of relations that can be used to describe a term related to other terms in the RDD specification.

As it has been shown in the previous section, these groups of relationships take different semantic points of view. The *Genealogy*, *Types* and *Membership of Sets* groups comprise relationships with semantics almost equivalent to RDF/S and OWL ones. The semantic equivalences have been deduced from RDD, RDF/S and OWL specifications.

The relations in this groups that can be mapped to RDF/S are presented in the upper part of Table 2. There is also a short description and the equivalent RDF property used to map them in RDDOnto. Only the RDD relations with an equivalent property in RDF/S are mapped at this level, i.e. *IsTypeOf*, *IsA*, *HasDomain* and *HasRange*. The

other relations have associated semantics that do not have equivalence in RDF/S. Therefore, if the mapping is restricted to the possibilities provided by RDF/S, then we get an incomplete ontology, i.e. it does not capture all the available semantics of RDD. However, on top of RDF/S, more advanced restriction building tools, like OWL, have been developed.

Using OWL ontology building blocks, some of the previously unmapped RDD relations can be mapped to the RDD ontology. In bottom part of Table 2 they are presented together with a short description and the equivalent OWL property used to map them in RDDOnto. With OWL almost all relationships can be mapped.

Only *Is* and *IsPartOf* relations do not have equivalents in OWL. Therefore, new properties in the RDDOnto namespace have been created to map them. Another alternative is to reuse other ontologies, as it has been done with Dublin Core. In this case mereological (*IsPartOf*) and quality (*Is*) notions are needed. For instance, they can be reused from the DOLCE [11] foundational ontology. For *IsPartOf* the equivalent is dolce:part-of and for *Is* it is dolce:has-quality.

Table 2. Mappings for relationships in the *Genealogy*, *Types* and *Membership of Sets* groups.

RDD relation	Short description	RDF
IsTypeOf	Builds the hierarchy of term types	rdfs:subClassOf rdfs:subPropertyOf
IsA	Relates an instance term to its type	rdf:type
HasDomain	Defines the source term type for relations	rdf:domain
HasRange	Defines the target term type for relations	rdf:range
IsMemberOf	The RelatingTerm from Member to Set	rdfs:member
RDD relation	**Short description**	**OWL**
Is	Relates resources to ascribed qualities	rddo:hasQuality
IsEquivalentTo	Relates two equivalent terms	owl:equivalentClass owl:equivalentProperty owl:sameIndividualAs
IsOpposedTo	Relates two opposite terms	owl:complementOf
IsPartOf	Relates a terms that is part of another term	rddo:isPartOf
IsAllowedValueOf	Relates allowed values to a type term	Inverse of owl:oneOf
HasType	The RelatingTerm from Archetype to Type	Inverse of rdfs:subClassOf rdfs:subPropertyOf
IsReciprocalOf	For relation terms defines the relation term that captures the inverse relation	owl:inverseOf

For the rest of the relationship groups, a part from *Genealogy*, there are no equivalent relations in the RDF/S plus OWL domain. This is due to the fact that these relationships are based on different kinds of semantics than those used in RDF/S and OWL. Therefore, the approach is to map them to new properties in the “rddo” namespace.

To conclude the mappings, it is also necessary to map RDD terms to Web ontology concepts. The previous mappings only cover the attributes that relate them. This has been postponed until now because Web ontology languages discern the RDD terms into three kinds: classes, properties and instances. The distinction is not made in RDD but it can be deduced from the term attributes.

If the term *Relationships* attribute includes *HasDomain* or *HasRange* relationships, it is clear that this terms must be mapped to a rdf:Property. This is a necessary and

sufficient condition because all terms referring to relations have at least one of this relationships.

Otherwise, the term is a class or an instance. It will be mapped to rdfs:Class if it has a *IsTypeOf* relationship or if there is no *IsA* relationship. If there is an *IsA* relationship but not *IsTypeOf* relationship, then it will be mapped to an instance, i.e. rdf:Description. It can be noted that it is possible to have a term that has both *IsTypeOf* and *IsA* relationships that is mapped to rdfs:Class. Therefore, as specified in the OWL Overview [12], the concrete OWL ontology produced is an OWL Full one.

3. Implementation

The RDD to RDF/S and OWL mappings that have been established in Table 1 and Table 2 have been implemented in the RDDOntoParser [13]. It is a Java implementation of these mappings using regular expressions [14]. Regular expressions are used to define patterns that detect the RDD part of the mappings. When patterns match, the corresponding RDF is generated in order to build RDDOnto.

Finally, once attributes have been mapped, they are used to discern the processed term as an rdfs:Class, a rdf:Property or an instance, rdf:Description. The input of the RDDOntoParser is a plain text version of Table 3 - Standardized Terms of the RDD standard [17]. The output constitutes the RDDOnto Web ontology [15]. For the other relationships a direct mapping to a new property with the same name in "rddo" namespace is implemented.

However, these relationships do not remain isolated in the resulting ontology. As all RDD terms are defined using RDD, relating terms are defined using relationships in the *Genealogy* group. Therefore, RDDOnto includes information about domain and range restrictions, relationships hierarchical organisation, etc.

4. Checking RDD with RDDOnto

During the ontology development, ontology tools facilitated the detection of integrity and consistency problems in RDD. There were many references to undefined references and inconsistencies between different parts of the standard. Some of these initial problems were communicated to the MPEG-21 RDD working group and the RDDOnto development process led to a revision [16] of the then recently published RDD ISO/IEC standard [17].

First of all, there were some inconsistencies between the textual RDD terms definitions and a figure showing the hierarchy tree of RDD act types. These inconsistencies were detected by comparing the figure included in the standard with a drawing of the Act hierarchy generated automatically from RDDOnto using the Protégé [18] ontology editor and the OntoViz [19] ontology visualisation plug-in.

However, the more important problems were related to the integrity issues of the standard. Some of the relationships and terms that were used in the terms definitions were not defined in it. Consequently, they have been added to RDDOnto, e.g. *HasCoChangedResource*, *icoInteractor*, *IsInteractorInContext*, etc. The integrity checks were performed with the help of the OWL validator vOWLidator [20].

Another testing facility once mapped to an OWL ontology is the consistency check provided by Description Logic (DL) [21] reasoners. OWL is a Description Logic so DL

reasoners can be directly used in order to reason with OWL ontologies. The only limitation is that reasoners only deal with two of the three OWL sublanguages, i.e. OWL DL and OWL Lite but not OWL Full. As it has been said, RDDOnto is OWL Full so we have to take away some of the mapped constraints that make it Full prior to feeding it into the DL reasoner. This has been done deactivating some of the mappings in the RDDOntoParser and with the further assistance of Protégé combined with the Racer DL reasoner. The more important feature that has been deactivated is the "IsA" to "rdf:type" mapping in order to avoid OWL Classes or Properties that are instances of other classes.

The interesting thing has been that, after making RDDOnto an OWL DL ontology, we have detected 320 inconsistencies in it. All of them are due to inconsistencies between the classes and properties hierarchies. The consequence is that many property domains and ranges are inconsistent with the domains and ranges of the corresponding superproperties. For instance, the property *IsAgentActingOn* has domain *Agent*. The direct superproperty *IsRelativeOf* has domain *Relative* but *Relative* is not a superclass of *Agent* so there is an inconsistency in the *IsAgentActingOn* domain.

These results of our ontological analysis of RDD have been submitted to the MPEG standardisation group [22] and its discussion has started a process to revise the standard in order to fix these problems.

5. Using RDDOnto

As it has been introduced, to have RDD formalised as an OWL ontology provides many advantages. The following sections describe some of them. First, there are the integration facilities provided by web ontologies that are used to integrate RDD in OWL form with other parts of MPEG-21, which are also mapped to OWL. Then, once in this integrated ontological framework, ontology-enabled tools like semantic query engines and DL reasoners facilitate the implementation of MPEG-21 tools.

5.1. Ontological Framework for Integration with MPEG-21 REL

The rights statements representation part of MPEG-21 is composed of the RDD, which defines the terms as it has been shown, but it also includes the Rights Expression Language (REL) [3]. The easiest way of explaining this is through a simile: the RDD provides the definition of the words while the REL provides a language to put these words together in order to build statements.

However, this intended complementarity is difficult to put into practice from the MPEG-21 standard specifications of REL and RDD. While the RDD is defined as an ontology, although not using a formal ontology language as it has been shown, REL is defined on the basis of a set of XML Schemas. This makes the integration between them very tricky.

Our approach has been to take profit from the integration facilities provided by web ontologies. The REL XML Schemas have been also mapped to OWL and then easily integrated with RDDOnto using the OWL semantic relations for equivalence and inclusion: *subClassOf*, *subPropertyOf*, *equivalentClass*, *equivalentProperty*, *sameIndividualAs*, etc. In order to map the XML Schemas to OWL and XML instances to RDF, the XSD2OWL and XML2RDF mappings [23] have been applied. The former is a generic mapping from XML Schemas to OWL ontologies, which has been also

applied to map another REL called ODRL to OWL ontologies [24]. The later maps XML instances, e.g. MPEG-21 licenses, to RDF taking into account the previous mappings from the XML Schemas used by the instance to their corresponding OWL ontologies. Thus, we get RDF versions of the licenses that are semantics-aware, i.e. they are connected to the ontologies that formalise the terms they use.

5.2. Semantic Query

Once the REL and the RDD were integrated, it was possible to develop ontology-enabled applications that take profit from their formal semantics. This has been used to implement MPEG-21 licenses management tools. For instance, the acts taxonomy in MPEG-21 RDD, see Figure 1, can be seamlessly integrated in order to facilitate license-checking implementation. Consider the scenario: we want to check if our set of licenses authorises us to uninstall a licensed program.

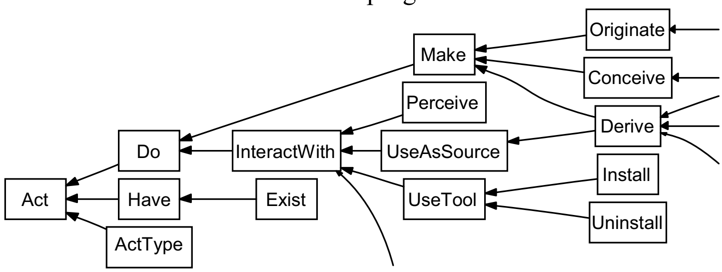

Figure 1. Portion of the acts taxonomy in MPEG-21 RDD.

If we use a purely syntactic approach like XPath over MPEG-21 XML licenses, there must be a path to look for licenses that grant the *uninstall* act, e.g. “//r:license/r:grant/mx:uninstall”. Moreover, as it is shown in the taxonomy, the *usetool* act is a generalisation of the *uninstall* act. Therefore, we must also check for licenses that grant us *usetool*, e.g “//r:license/r:grant/mx:uninstall”. And successively, we should check for *interactwith*, *do* and *act*. All this must be done programmatically, the XPath queries are generated after we check the RDD ontology.

However, if we use semantic queries, the existence of a license that grants any of the acts that generalise *uninstall* implies that the license also states that the *uninstall* act is also granted. This is so because, by inference, the presence of the fact that relates the license to the granted act implies all the facts that relate the license to all the acts that specialise this act.

Therefore, it would suffice to check the semantic query “//r:license/r:grant/mx:uninstall”. If any of the more general acts were granted, it would match. For instance, the XML fragment “/r:license/r:grant/dd:usetool” implies the fragments “/r:license/r:grant/dd:install” and “/r:license/r:grant/dd:uninstall”.

5.3. Usage against License Checking Using DL Reasoners

There are other application development facilities more sophisticated than the semantic queries benefits shown before. One of the most promising tool is Description Logics (DL). OWL is based on DL so it can be directly fed into DL classifiers. Classifiers are specialised logic reasoners that guarantee computable results. DL classifiers are used with RDDOnto in order to automatically check IP uses against the use patterns

specified in IP agreements or offers. This facilitates checking if a particular use is allowed in the context of a set of licenses or finding an offer that enables it, once an agreement is reached.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. In order to do that the following steps are followed:

1. First of all, the usage event that is going to be checked is modelled as instance data using RDDOnto and the REL ontology. For instance: "USER1 is trying to access a given video stream from a given streaming server at 9:30:10 UTC on 2005-04-10". The streaming server implements digital rights management so it inquires the license manager if the current usage instance is permitted. In order to do that, the streamer models this usage and sends it to the license manager, e.g. as a RDF/XML serialisation.
2. The license manager contains licenses also modelled using the RDD and REL ontologies. However, they are modelled as classes. These licenses define usage patterns and the conditions to be fulfilled in order to be authorised. When the pattern refers to a particular instance, e.g. the particular video stream on the previous example, the license class is defined by OWL *hasValue* constraints. This kind of constraints defines a class of instances that are related by a given property to a particular instance. When the constraints on the usage pattern are more general, e.g. a set of users of which the USER1 in the example is a member, OWL constraints like *allValuesFrom* or *someValuesFrom* are used. They are defined for the given property and to the corresponding class, e.g. an enumerated class containing USER1.
3. The license manager checks if there is any license that grants a usage pattern that subsumes the usage instance. This can be performed easily and efficiently using a DL classifier. However, there are some problems that should we resolved before. The usage patterns may define time intervals that should be tested against the usage time point. In order to check if the time point is included in the time interval, we must use a DL classifier capable of dealing with custom datatypes reasoning [25]. Then, the time interval is translated to a real interval (pointInTime.$\geq[20050401]^{real} \wedge \leq[20060401]^{real}$) and the time point to a real (pointInTime.$=[20050410.093010]^{real}$).
4. After applying the previous adaptations, subsumption is computed. The usage might be classified in one or more usage patterns. In this case it is tested if the usage pattern is contained in a grant or and offer. In the first case, the condition is checked and if it is satisfied the license manager tells the streaming server that the use is authorised. Otherwise, the use is not authorised. In the second case, the offer that has been found may be accepted or negotiated in order to achieve the usage.

6. Conclusions and Future Work

We have presented RDDOnto, an ontology for the MPEG-21 RDD. Its added value over other initiatives to implement rights data dictionaries is that it is based on applying an ontological approach. This is done by modelling the RDD standard using ontologies. Ontologies allow that a greater part of the standard is formalised and thus more easily available for implementation, verification, consistency checking, etc.

RDDOnto demonstrates the benefits of capturing the RDD semantics in a computer-aware formalisation. First of all it has been possible to analyse the standard integrity and consistency with the support of ontology-aware tools that facilitate this issue, discovering inconsistencies that are in the process of being fixed in the standard. Then, it has been possible to integrate RDD with another MPEG-21 standard part, the Rights Expression Language (REL), in a common ontological framework. This framework facilitates the implementation of MPEG-21 tools. We have shown our achievements using semantic query engines and Description Logic reasoners for license searching, validation and checking. The ontological approach has also made possible the development of advanced Digital Rights Management systems that integrate these tools in order to build semantic information systems [26,27] and intelligent agents for assisted rights negotiation [28].

The objective now is to take profit from the abstraction and integration facilities of formal ontologies in order to cope with the RDD standard problems. First of all, RDDOnto is being used in order to extend RDD capabilities in a consistent and more informed way. Some communities might find that there are some unsatisfied requirements in the current RDD. This is completely normal as it is impossible to cope with all the requirements of communities as big as the ones that might be interested in the MPEG-21 standard.

The MPEG-21 RDD standard specifies mechanisms for standard extension. However, it is difficult to put these mechanisms into practise. The size of the standard makes it very complex for people outside the standardisation process to manipulate and extend it in order to satisfy their particular needs. This is why we have started to use RDDOnto as an assistance mechanism for RDD testing of new requirements. RDDOnto is used together with ontology rendering tools in order to navigate the RDD hierarchy of concepts, detect the part of it where the new concept might be situated and even produce a graphical drawing of it

Another future line is to exploit the integration possibilities of OWL in order to connect RDDOnto with more general Intellectual Property Ontologies, e.g. IPROnto [29], or rights data dictionaries of other rights expression languages like ODRL. The objective here is to build an ontology-based framework that allows integrating these initiatives, making them interoperable and enrich them with the possibilities offered by formal ontologies. This might lead to levels of interoperability that allow combining different RELs and RDDs in a totally uncoupled way.

Acknowledgements

This work has been partly supported by the Spanish administration (AgentWeb, TIC 2002-01336). The final part of the work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 program.

References

[1] Moving Picture Experts Group (MPEG) ISO/IEC/JTC1 SC29/WG11, http://www.chiariglione.org/mpeg/index.htm.

[2] Walle, R.v.d.: The MPEG-21 Book, John Wiley & Sons, Chichester, 2005.

[3] Wang, X.; DeMartini, T.; Wragg, B. and Paramasivam, M.: The MPEG-21 Rights Expression Language, IEEE Transactions on Multimedia 7(2), 2005.
[4] Rust, G. and Barlas, C.: The MPEG-21 Rights Data Dictionary, IEEE Transactions on Multimedia 7(2), 2005.
[5] Iannella, R.: Open Digital Rights Language (ODRL), Version 1.1. Technical report, World Wide Web Consortium, 2002
[6] Dean, M. and Schreiber, G. (eds.): OWL Web Ontology Language Reference, W3C Recommendation, 2004. http://www.w3.org/TR/owl-ref.
[7] Lassila, O. and Swick, R.R. (eds.): Resource Description Framework (RDF), Syntax Specification, W3C Recommendation, 2004. http://www.w3.org/TR/rdf-syntax-grammar.
[8] Brickley, D. and Guha, R.V. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 2004. http://www.w3.org/TR/rdf-schema.
[9] Sowa, J.F. (ed.): Principles of Semantic Networks: Explorations in the Representation of Knowledge. Morgan Kaufmann Publishers, San Mateo, CA, 1991.
[10] Dublin Core Metadata Element Set, Version 1.1: Reference Description, http://dublincore.org/documents/dces.
[11] Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N. And Oltramari A.: Ontology Library, v1.0. IST Project WorderWeb, Deliverable 18, 2003. http://wonderweb.semanticweb.org/deliverables/D18.shtml.
[12] McGuinness, D. L. and van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation, 2004. http://www.w3.org/TR/owl-features.
[13] García, R.; Delgado, J.; Rodríguez, E.; Llorente, S. and Gallego I.: RDDOnto, Rights Data Dictionary Ontology Version 2, ISO/IECJTC1/SC29/WG11/M10423, December 2003.
[14] Java 2 Platform Std. v1.4.2, Package java.util.regex, http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/package-summary.html.
[15] MPEG-21 REL and RDD Ontologies, http://dmag.upf.edu/ontologies/mpeg21ontos.
[16] Barlas, C. & Rust, G.: Rights Data Dictionary, Technical Corrigendum 1, ISO/IEC JTC 1/SC 29/WG 11, 2005.
[17] ISO/IEC 21000-6:2004, MPEG-21 Part 6: Rights Data Dictionary (RDD), ISO/IEC JTC 1/SC 29, 2004.
[18] Protégé Project, http://protege.stanford.edu.
[19] OntoViz, Protégé ontology visualisation plug-in, http://protege.stanford.edu/plugins/ontoviz/ontoviz.html.
[20] vOWLidator, http://projects.semwebcentral.org/projects/vowlidator
[21] Pan, J.Z.: Description Logics: Reasoning Support for the Semantic Web. PhD thesis, School of Computer Science, The University of Manchester, 2004.
[22] García, R.; Delgado, J. and Rodríguez: Ontological Analysis of the MPEG-21 Rights Data Dictionary (RDD). ISO/IECJTC1/SC29/WG11/M12495, October 2005
[23] Gil, R.; García, R. and Jaime Delgado: An interoperable framework for IPR using web ontologies. Legal Ontologies and Artificial Intelligence Techniques, LOAIT'05. IAAIL Workshop Series, Wolf Legal Publishers, pp. 135-148, 2005.
[24] García, R.; Gil, R.; Gallego, I. and Jaime Delgado: Formalising ODRL Semantics using Web Ontologies. Open Digital Rights Language Workshop, 2005.
[25] Pan, J.Z. and Horrocks, I.: OWL-Eu: Adding Customised Datatypes into OWL. In Euzenat, J. and Gómez-Pérez, A. (ed.): Proc. of Second European Semantic Web Conference (ESWC 2005). Springer-Verlag, LNCS Vol. 3532, pp. 153-166, 2005.
[26] García, R.; Gil, R. and Delgado, J.: Intellectual Property Rights Management using a Semantic Web Information System. In R. Meersman and Z. Tari (ed.): OTM Confederated Intera tional Conferences, CoopIS, DOA, and ODBASE '04. Springer-Verlag, LNCS Vol. 3291, pp. 689-704, 2004.
[27] García, R. and Delgado, J.: Brokerage of Intellectual Property Rights in the Semantic Web. In Cruz, I.F.; Decker, S.; Euzenat, J. and McGuinness, D.L. (ed.): Proc. Semantic Web Working Symposium. Stanford University, California, pp. 245-260, 2001.
[28] Gil, R.; García, R. and Delgado, J.: Delivery context negotiated by mobile agents using CC/PP. In Mobile Agents for Telecom Applications, (MATA'03). Springer-Verlag, pp. 99-110, 2003.
[29] Delgado, J.; Gallego, I.; Garcia, R. and Gil, R.: IPROnto: Intellectual Property Rights Ontology. In Ext. Poster Abstracts in Proc. International Semantic Web Conference, 2003. http://dmag.upf.edu/ontologies/ipronto

Legal Knowledge and Information Systems
M.-F. Moens and P. Spyns (Eds.)
IOS Press, 2005

Using Legal Definitions to Increase the Accessibility of Legal Documents

Laurens Mommers and Wim Voermans
Faculty of Law, Department of State and Administrative Law,
Leiden University, The Netherlands

Abstract. Access to legal documents has been hampered by the lack of attention for specific user groups accessing such documents. In this article, we focus on one of these user groups (legal professionals), who can benefit from specific types of cross-lingual information retrieval for, *e.g.*, comparative law research. We propose to use legal definitions as anchor points in legal documents. Through the body of EU legislation, these anchor points can support a network of concepts between different jurisdictions. A model is presented containing the different entity types and relations types for building such a network, which can be implemented in the WordNet architecture.

1. Introduction

Accessibility of legal documents can be supported in different manners, for instance by hyperlinking related documents, summarizing their content, highlighting their structure, supporting query formulation, full-text search, or thesaurus-based search. A suitable selection of support methods depends on the target user group, and the type of use they make of the document set. Laymen will generally need more support in finding the information they are looking for than legal professionals, and getting a rough idea of the content of a series of documents imposes other demands on access methods than finding a specific legal decision. The recognition of this rather obvious fact is seen nowadays in the differentiation of web services with a legal nature. The Dutch government not only offers a web-site with all valid legislation (www.overheid.nl/wetten), but also a 'product catalogue' (overheidsloket.overheid.nl), which makes accessible the services and products that emanate from valid legislation.

The LOIS project (lexical ontologies for legal information sharing) was initiated to increase the accessibility to legal documents (a) between different EU languages, (b) for persons with limited knowledge of the professional vocabularies used in law, and (c) for legal professionals. For this purpose, a large multi-lingual WordNet was proposed, supporting monolingual and interlingual information retrieval. The end product of the LOIS project will contain around 5,000 concepts (so-called 'synsets') per language [1]. These concepts are linked to each other (within and between languages) in meaningful ways, *e.g.*, by hierarchical relations and conceptual equivalence. In this article, we narrow down our scope to the use of legal concepts to support the accessibility of legal documents. The main question is: how can legal definitions in legislation be used to support the accessibility of legal documents?

In order to give an answer to the main question, in section 2, we will explain how the accessibility of legal documents can be increased through particular uses of legal

concepts. In section 3, we will elaborate on the different ways in which legal concepts are defined by legislators. Section 4 is devoted to a model for conceptual representation using legal definitions. Section 5 contains our conclusions.[1]

2. Supporting Accessibility

Information retrieval is often based on the occurrence of terms in a query and in documents retrieved. However, the quality of search results depends on the degree to which the *meaning* of certain terms can be determined. The fact whether I am searching for documents on the *legal* term ‘agreement’ or on the *lexical* term ‘agreement’ is relevant to determining suitable search results (a text may be about, for instance, a political agreement, which probably makes it useless to a lawyer). This was the main reason to introduce legal terms and their (legal) definitions in the legal WordNet for the LOIS project, although the WordNet framework by default only features lexical definitions of terms. In the legal domain, term definitions may deviate from lexical definitions.

2.1. Using Legal Definitions for Retrieval Purposes

The lexical origination of the WordNet architecture has several consequences for the accommodation of legal terms and definitions. First, definitions ought to have a lexical nature, *i.e.*, they need to be explanatory rather than normative. Second, synonyms within a synset have an equivalent status (there are no preferred terms, such as in the case of a thesaurus). Third, relations between synsets have a lexical nature, *i.e.*, they generally fit in with lexically-oriented relations, such as hypernymy and hyponymy.

In order to use legal definitions in a WordNet in a meaningful manner, we thus have to stretch the boundaries of the original WordNet architecture [2]. First, legal definitions have a normative nature, *i.e.*, they establish meanings of legal terms independent of their generally perceived (lexical) meanings. Second, the use and definitions of terms in the legal domain determine whether there are true synonyms. In most cases, synonymy does not occur, simply because legal terms have normative definitions, and there is not much use in coining two different terms with identical normative definitions. Third, using legal terms and their definitions requires the employment of other relations than lexical ones. Although lexical relations can still apply in many cases, legal relations are needed for some applications of a legal WordNet.

Insofar as lexical WordNet relations apply to the legal concepts in the LOIS WordNet, a legal system, rather than a lexical system, determines which legal concepts are related to which other legal concepts. For instance, hyponomy and hypernomy will apply when they fit in with the ordering of concepts within a legal system. The principles of hyponomy and hypernomy will remain the same, although their actual use depends on the relation between concepts as dictated by a legal system. Insofar as legal relations are introduced, these reflect typical relation types between legal concepts that are not found in the ‘normal’ lexical domain.

2.2. Using Legal Definitions in Applications

There is an increasing interest in comparative law research, both for the comparison of national legal systems, and for the comparison of implementations of EU legislation in Member States. This interest is caused by, among other things, international trade rela-

[1] We would like to thank Sara Castagnoli and three anonymous referents for their valuable comments to a previous version of this paper.

tions and the need to implement EU legislation properly. It calls for ways to access documents from other legal systems, which are often written in different languages (Dutch vs. English legislation), or at least employ different legal vocabularies (Dutch vs. Belgian legislation).

What are the concrete opportunities offered by using legal definitions in a WordNet? First, relations in the WordNet can be used to follow a legal term to its 'equivalent' in a different language. Comparing national definitions of terms from European directives – and the context in which they are used – can be helpful for persons who are responsible for implementing those directives in a different national legal system. Second, if implementation has taken place, comparative law research can be of use in explaining the meaning of terms emanating from European legislation. For instance, telecommunication enterprises may – for various reasons - want to know how telecommunication directives have been implemented in other Member States and in which codes, Acts or any other legal documents these implementations have been enshrined – be it correctly or incorrectly. And a market regulator, like the Dutch telecommunications regulator OPTA, may well benefit from the explanation or interpretation of certain implemented terms by regulatory authorities or courts in different countries.

The scope of term definitions from EU legislation may be further determined by extending the search to documents other than just EU and national legislation. Court decisions, official publications and legal literature can be added to the document set. If there are no explicit links between an EU legislative document and a different document, a link between the two documents may be established through the occurrence of identical terms. This is how information retrieval usually works. The results of searches can be restricted by disambiguating – as much as possible – the meanings of identical terms. If a term such as 'employee' has different definitions in different legal sources, those different meanings should help us in finding only relevant documents, because the scope of a definition is often explicitly determined in legislative texts.

What does this mean for a user who is looking for documents relevant to a certain legal concept? First, the user can disambiguate the term. This is attained by presenting the user with the different definitions there are for a term, and let the user select the relevant definitions. Second, the retrieval system may perform the disambiguation. This can only be attained if the query contains sufficient terms to determine a context automatically. Third, the disambiguation may take place interactively, by starting the search with the user query and presenting all documents that contain the term(s) entered. The scope is then narrowed down by the user, who selects relevant documents. This may, in turn, provide valuable information to the retrieval software.

In a retrieval application, using a multilingual WordNet based on legally equivalent concepts derived from EU legislation with accompanying scope information can substantially support both the recall (across EU languages) and the precision (through disambiguating the meaning of homonyms).[2]

3. Defining Legal Concepts

Legal terms often have (partially) explicit definitions associated with them. This is especially the case in statute law, in which certain sections are devoted to making explicit the meaning of certain terms within the context of the statute under scrutiny. The types of definitions found in legislation are described below. Such definitions are easy to

[2] In addition to a demonstrator application (see www.loisproject.org), the LOIS WordNet will be integrated in existing retrieval applications of several companies as a part of the LOIS project.

identify, as they are often introduced by phrases such as: 'Within this statute, *x* means *y*', where *x* stands for the term to be defined, and *y* for the definition. They are often found in approximately the same place in a statute, nearly in the beginning. Examples of this can be found in European legislation as well; in European directives, definitions are found mostly in article 1 or 2, directly after the considerations.

3.1. Definition Types in Law

Definitions in legislation aim to clarify the meaning of terms in order to prevent vagueness and ambiguity. They have a function and status different from definitions in an ordinary text. As a part of the body of a legislative text, legislative definitions are *norms;* they are legally binding on whomever interprets, enforces or implements legislation. This sets legislative definitions apart from lexical definitions. Because of this normative nature of legislative definitions they should only be used when strictly necessary. According to article 14 of the Interinstitutional agreement of 22 December 1998 on common guidelines for the quality of drafting of Community legislation – a draftsman's handbook for European legislators – only ambiguous terms are to be defined in a way that differs from the lexical meaning. Many other countries and drafting cultures share this restrictive rule; definitional proliferation creates rather than limits interpretation problems.

According to Eijlander and Voermans [3], there are basically four techniques to define a term in a legislative text:

1. *No definition* (this is the technique whereby the legislator does not define a term at all, but relies on the everyday, lexical meaning, or even specialized language meaning – jargon – of a term. This technique is the most common technique and the preferred one. Only if a term is ambiguous, a legislative definition may be considered);
2. *Definition by context* (a given term may be ambiguous in itself, but put into the context of a given legislative text, its meaning is clear and precise);
3. *Definition* (whereby a – partial or whole - new meaning is given to a term). There are four definition types:
    a. *Generalizations* (defining by giving a general description. For instance in article 3, par. 3, of Directive 91/439/EEC 'power-driven vehicle' is defined as 'any self-propelled vehicle running on the road under its own power, other than a rail-borne vehicle');
    b. *Specifications* (defining by listing the elements that constitute the concept and hence its meaning. *E.g.*, article 2, par. 1 of Directive 2004/17/EC defines 'central purchasing body' as a 'contracting authority (…) which: acquires supplies and/or services intended for contracting entities; or: awards public contracts or concludes framework agreements for works, supplies or services intended for contracting entities.');
    c. *Recursive definitions* (defining by listing along the lines of a *decreasing* set of elements constituting the concept; stepping down as it were. Recursive definitions are quite rare, but an example of this technique is found in article 2, par. 2 of Directive 2004/38 EC where 'Family member' means: '(a) the spouse; (b) the partner with whom the Union citizen has contracted a registered partnership [...]; (c) the direct descendants who are under the age of 21 or are dependants and those of the spouse or partner as defined in point (b); (d) the de-

pendent direct relatives in the ascending line and those of the spouse or partner as defined in point (b)'); or

d. *Abbreviations* (defining by giving an practicable abbreviation for a long or complicated term. E.g. a quite common example derived fro Dutch legislation: 'Our Minister' shall mean 'Our Minister of Agriculture, Nature and Food Quality').

4. *Definitions by reference* (whereby a term is defined by way of reference. Example: the 'x' as meant in 'y'. *E.g.*, article 2, par. 1, point c, of Directive 2003/30/EC defines 'other renewable fuels' as 'renewable fuels (…) as defined in Directive 2001/77/EC(8) (…)'.

Thornton [4] uses a somewhat different angle of approach and distinguishes two cumulative functions of definitions: the avoidance of ambiguity and the avoidance of repetition. Rather than discerning different definition techniques, Thornton is interested in the relation between legislative definitions and the lexical meaning of terms. To the notion of different definition techniques and definition types, he adds the element of *delimiting*, *extending* and *narrowing* definitions. Delimiting, extending and narrowing definitions are typically used to clarify the meaning of a certain term by stripping elements from a known definition or by adding new elements.

Delimiting definitions, for instance, refer to lexical meaning, but remove some of the vagueness. For instance, in Directive 2000/12 EC, 'credit institution' is defined as 'an undertaking whose business is to receive deposits or other repayable funds from the public and to grant credits for its own account', which is a fair representation of the lexical meaning of the concept, but removes most vagueness. Extending definitions expand the scope of a term with respect to its common meaning. An example of such a definition describes 'person' as including 'a corporation sole and also a body of persons whether corporate or unincorporate' [4]. Narrowing definitions limit the scope of a term with respect to its common meaning. For instance, in Directive 2000/12 EC, 'authorization' means 'an instrument issued in any form by the authorities by which the right to carry on the business of a credit institution is granted', which obviously limits the scope of the lexical meaning of 'authorization'.

With respect to these definition techniques and definition types, simplifications can be made in order to support the current research. With respect to definition techniques: providing no definition and defining by context do not yield explicit definitions in writing, and are therefore left out of consideration. Defining by generalization and by specification yield explicit definitions (enumerations of application conditions, and enumerations of entities/entity types, respectively). Defining by recursion can yield a definition of either type. Defining by abbreviation create synonymy between terms, and defining by reference constitutes a referral to a different definition – which can be of any type. In the context of this article, Thornton's [4] notion of delimiting, extending and narrowing definitions is especially useful in case of relating lexical defintions and legal definitions to each other. If we can determine what type of legal definition is used for a term compared to its lexical counterpart, this gives important information regarding possible WordNet relations to be established between the corresponding synsets.

3.2. Community Legislation: A Source of Conceptual Equivalence

In European Community legislation, a unique situation is created regarding legal meaning. All language versions of legislative and case law documents are deemed to be equivalent (*cf* EEC Council, Regulation N° 1, determining the languages to be used by

the European Economic Community, *Official Journal* 1958, p. 385-386). Thus, for instance, the legal effect of the Dutch version of a European directive is deemed identical to the legal effect of the English or Greek version of that directive. Although there can be objections to this principle, relating to practical translation difficulties and theoretical meaning discussions, the effect of the principle is that there is a common basis for assessing meanings of legal concepts in different EU languages. And although many of these legal concepts remain inherently ‘European’, because they leave no direct traces in national legislation, there is a category of documents that establishes an explicit link between Community legislation and national legislation. European *directives* provide measures that should be implemented in national legislation. For this purpose, any directive contains a series of norms. One of these is a definition article, containing a list of concept definitions.

Member States can either choose to implement these definitions literally (the so-called copy-out-technique, which is the preferred and most commonly used technique throughout Europe) (*cf.* [6], p. 15-16 and [7], directive 56) or they can opt for a different definition, for multiple definitions, or no definition at all (the so-called elaboration-technique) (*cf* [6], p. 16). The elaboration technique is used to create a better fit between a directive and the terminology or structure of national legislation. Of course transposition of directive terms by way of elaboration needs to remain within the preconditions set by the directive.

Thus, in a number of cases, an explicit implementation relation can be established between concepts in directives and concepts in national legislation. This is not only possible if the copy-out technique of transposition has been used, but in some cases even when the elaboration technique method was employed. Elaboration does, indeed, in some cases leave a traceable track in national legislation. This trace may for instance be found in an explanatory memorandum. In The Netherlands, for instance, an explanatory memorandum accompanying a Dutch Act, decree or regulation implementing an EU directive needs to contain a transposition table, relating the provisions in the EU directive to the national regulation ([7], directive 344).

The implementation relation, in itself, does not say anything about the way in which a concept is implemented. It only says *that* a concept has been implemented. The implementation relation can be complemented by a relation stating the nature of the link between the original concept and the implemented concept(s). For instance, if the definition of the national legal concept is identical to the Community legislative concept, an equivalence relation can be established. If the definitions are almost identical, a near equivalence relation is assumed. If the national concept has a definition more specific than the Community concept, the former is a narrower term of the latter. If the national concept has a more general definition than the Community concept, the former is a broader term of the latter.

One of the areas in which EU legislation plays a prominent role, is in the regulation of telecommunications (nowadays called ‘electronic communication’). Many examples of implemented definitions can be found here. For instance, in Directive 2002/22/EC, we read: ‘“public telephone network” means an electronic communications network which is used to provide publicly available telephone services; it supports the transfer between network termination points of speech communications, and also other forms of communication, such as facsimile and data’. A (supposedly) identical definition is available for all other official EU languages, among which is Dutch. The implementation of the concept ‘openbaar telefoonnetwerk’ is identical to the defi-

nition in the Directive (*cf.* art. 1.1 sub w Dutch Telecommunications Law). Clearly, the 'copy out' technique has been used for this particular implementation.

4. A Model of Cross-Lingual Legal Concept Comparison

On the basis of the observations in this paper, a model was designed for linking concepts from different legal systems in various languages. This model is based on the following assumptions. First, the meaning of legal concepts is for the greater part established in authoritative legal documents. Such documents constitute the formal sources of law: legislation, case law and customary law (additionally, doctrine is sometimes regarded as a formal source of law - the authority of doctrine is often questioned; however, it plays an important role in defining legal concepts). Such legal documents contain terms, some of which explicitly refer to legal definitions, whereas the meaning of other ones is established on the basis of everyday or contextual use. For explicit definitions, assembling application conditions is relatively easy. Sometimes, additional conditions have to be assembled from other sources; *e.g.*, different parts of legislation, and discussions in authoritative case law or doctrine.

If there are not sufficient explicit conditions in a definition, or a definition is lacking, different interpretation methods can be used, such as the grammatical, systematic, comparative law and teleological interpretation methods (for an overview of such methods, *cf.* [7]). In case of explaining the meaning of terms, these methods essentially support making explicit the application conditions that were left implicit, *e.g.*, at the time of drafting the legislative text. 'Implicit' refers to the legislative text itself. Application conditions may be found formulated explicitly in the parliamentary history of a proposal for legislation, or in a judicial opinion.

A term with an assigned meaning (either a legal definition, or an everyday or contextual definition) is a *concept*. Thus, legal documents contain terms, and terms refer to concepts, which on their turn are constructed from definitions or definition parts found in legal documents. The consequence of using legal definitions is that one term may be defined in multiple ways: different legal definitions may occur for a term such as 'consumer', and for the same term, a lexical definition may be provided. The term 'consumer' may have a meaning in agricultural legislation different from its meaning in consumer protection law.

4.1. Comparative Terminology

Castagnoli [8] describes the task of comparative terminology as the determination of the degree of equivalence between the concepts associated with related terms in different languages (also *cf.* [9]). She distinguishes three levels of equivalence: *perfect equivalence* occurs if the concepts expressed by the terms in the source language and the target language are identical. *Partial equivalence* occurs if the concepts expressed by the terms in the source language and the target language are more or less equivalent. *Non-equivalence* occurs if a concept in the source language has no equivalent in the target language. The second equivalence level, partial equivalence, has different manifestations. First, it can occur through different lexicalisations (the source language has one term referring to a certain concept, whereas the target language has multiple terms for two similar concepts). Second, two concepts in the target language may represent a distinction between meanings that is not found in the source language, which only has one term and a corresponding concept. Third, two terms (in the source language resp. in the target language) may be lexically related, but the concepts they refer to are dif-

ferent. Table 1 is based on Castagnoli's (2005) work, and also provides her examples of each of the equivalence types.

Table 1. Equivalence types.

equivalence degree type	**equivalence degree subtype**	**example (it = Italian, en = English)**
perfect equivalence		it interesse = en interest
partial equivalence	different lexicalisations	it bene mobile ≈ en chattel, personal property
	partial equivalence	it tassa/imposta ≈ en tax
	lexical equivalence, conceptual difference	it contratto ≈ en contract
non-equivalence		it negozio giuridico ≠ en legal act

In addition, Castagnoli distinguishes between translation equivalence and functional equivalence. This distinction is relevant to, *e.g.*, country-specific institutions: the Italian term 'regione' refers to a concept which has a functional equivalence in England, namely the concept referred to by 'county', and translation equivalence denoted by the term 'region'. Both the different degrees of equivalence and the types of equivalence are a clear illustration of the problems that may be encountered if we attempt to link concepts between different languages. There is often no such thing as *the* translation of a term; there is only a more or less proper translation of a term within a certain context. However, the 'artificial' concepts in the legal domain provide support for making explicit differences and similarities between concepts.

Castagnoli's distinctions may be used to impose relations on legal terms in different languages. On their turn, these relations support the capabilities of a retrieval application. Obvious relation types to be introduced are: lexical equivalence (terms are 'literal translations' between languages, often occurring as a consequence of the common origination of European languages), conceptual equivalence (the concepts that terms refer to have identical application conditions), functional equivalence (the concepts that terms refer to play the same role in the two legal systems concerned). For conceptual and functional equivalence, a degree of equivalence can be used: if there is no equivalence, the conceptual or functional equivalence will not occur. If there is partial equivalence, this constitutes a characteristic of the equivalence relation, and if there is full equivalence, the same is valid. In a WordNet context, there is no second order description available, so relations cannot have characteristics. Therefore, we would end up with the following relation types: lexical equivalence, full conceptual equivalence, partial conceptual equivalence, full functional equivalence and partial functional equivalence. Lexical equivalence will be left out in the model in subsection 4.2, as we focus on the *meanings* of terms, not on their lexical appearance.

4.2. Overview of the Model

Relations among legal concepts may provide insight into the structure of legal systems. As such, they can facilitate retrieval of relevant, related information. With respect to Community directives, two types of relations are distinguished: structural and content relations. Structural relations reflect actual systemic connections between legal concepts; content relations reflect similarities or differences among the meanings of legal concepts. Structural relations needed in the current model are an 'implemented as' rela-

tion (a relation between two legal concepts, one of which is part of an EU directive, the other of which is part of a national regulation based on that directive), an 'implemented in' relation (a relation between two legal documents, one of which is an EU directive, the other of which is a national regulation based on that directive). Content relations are based on the discussion of comparative terminology in section 4.1. These relations include conceptual equivalence, functional equivalence and legal equivalence.

In table 2, an overview of entities and relation types of the model of legal definitions is provided. These entities and relations can be implemented in the WordNet framework. Currently, the 'implemented as' and 'implemented in' relations are part of the LOIS WordNet. Synsets based on all definition articles present in EU directives in section 15.20 (Consumers) of the EU analytical register have been added to the LOIS WordNet. In addition to this, a selection of synsets based on definitions in other EU directives has been added to the WordNet, totalling around 2,000 synsets for the legal part of the WordNet (excluding the implemented national legal concepts still to be connected to the EU concepts).

Table 2. Entities and relations.

entities and relations	**explanation**
legal document	document (pertaining to a formal source of law)
legal term	term present in a legal document
legal concept	term with 'attached' meaning, made explicit in the form of application conditions or an enumeration, originating from (a) legal document(s)
'implemented as' relation	relation between two legal concepts, one of which is part of an EU directive, the other of a national regulation based on that directive
'implemented in' relation	relation between two legal documents, one of which is an EU directive, the other a national regulation based on that directive
full conceptual equivalence	relation between two legal concepts: the application conditions (or the entities referred to) of the two concepts are equivalent
partial conceptual equivalence	relation between two legal concepts: the application conditions (or the entities referred to) of the two concepts are partly equivalent
full functional equivalence	relation between two legal concepts: the entities referred by the two concepts play an equivalent role
partial functional equivalence	relation between two legal concepts: the entities referred by the two concepts play a partly equivalent role
full legal equivalence	relation between concepts: full conceptual equivalence accompanied by legal recognition of this equivalence (such is the case with concepts from EU legislation in different languages)
partial legal equivalence	relation between concepts: partial conceptual equivalence accompanied by legal recognition of this equivalence (such is the case with certain concepts in EU legislation and their implementations in national regulations)

5. Conclusions

Legal definitions can provide valuable support in information retrieval applications. The LOIS WordNet contains lexical as well as legal definitions, and relations between them. The legal concepts in the WordNet may serve to support specific information retrieval applications useful to legal professionals. For instance, lawyers can look up implementations of EU legislation by following the implementation links between con-

cepts embedded in the WordNet, or they can perform comparative law research. The model proposed for the legal concepts part of the LOIS WordNet is based on the way in which authoritative legal documents specify the meaning of legal terms: partly by way of explicit definitions. These definitions determine the meaning of legal terms for a specific (legal) context. In the LOIS WordNet, the legal terms with accompanying definitions serve as *synsets* (sets of synonyms for specific terms, and the legal definition of these terms as so-called *glosses*). Relations between concepts can be established on the basis of certain legal connections, such as the relation between EU directives and their implementations in national legislation. By way of using the EU directives as ‘junctions’ in the WordNet, a relation is established between legal concepts in different languages.

This model of legal meaning assumes that there is no necessary relation between lexical and legal meaning. Legal concepts *may* be, for instance, specifications of lexical concepts, but the legislator may as well use, for any legal term, a meaning totally different meaning from the lexical meaning. As became clear from the different definition types elaborated on in subsection 3.1, the legislator has quite extensive tools in ‘manipulating’ or ‘creating’ meanings for terms. Therefore, traditional translation difficulties are also different for legal terms. On the one hand, differences between legal concepts may be larger than between lexical concepts, making translation more difficult. On the other hand, meaning differences are more often made explicit in authoritative legal documents, which makes translation easier. In the case of EU legislation, the situation is different: legal concepts in different languages are deemed equivalent.

EU legislation therefore plays a major role in enabling cross-lingual information retrieval in legal domains. It constitutes a source of conceptual and functional equivalence, but above all, it constitutes a source of legal equivalence. As EU directives have to be implemented in national legislation, the legal concepts from these directives leave their traces in national legislation, providing anchors for cross-lingual information retrieval. On the basis of Castagnoli’s (2005) observations, the equivalence relations can be further specified in terms of conceptual, functional and legal equivalence. In the case of EU legislation in different languages, relevant forms of equivalence can often be found (full or partial legal equivalence, based on the legal recognition of full or partial conceptual equivalence).

References

[1] L. Dini, D. Liebwald, L. Mommers, W. Peters, E. Schweighofer and W. Voermans, Cross-lingual legal information retrieval using a WordNet architecture, *Proceedings of ICAIL '05*, 2005, 163-167.

[2] C. Fellbaum (ed.), *WordNet. An Electronic Lexical Database*, MIT Press, 1998.

[3] P. Eijlander and W. Voermans, *Wetgevingsleer*, Boom Juridische uitgevers, 2000.

[4] G.C. Thornton, *Legislative Drafting* (4th edition), Butterworths, 1996

[5] Cabinet Office, *Transposition Guide; How to implement European Directives effectively*, London, 2005.

[6] Drafting Directives, *Aanwijzingen voor de regelgeving (Dutch Drafting Directives)*, Den Haag, 2005.

[7] H. Franken, *Encyclopedie van de rechtswetenschap*, Gouda Quint, 2003.

[8] S. Castagnoli, Da database terminologico a knowledge base integrata: il sito web 'Sicurezza, Salute e Ambiente nei luoghi di lavoro, communication presented at La Formazione in Terminologia: Portico di Romagna’, Forlì, April 29, 2005.

[9] R. Arntz, Confrontare, valutare, trasporre: metodi e problemi della traduzione giuridica, in: R. Arntz (ed.), La traduzione – Nuovi approcci tra teoria e pratica, Accademia Europea di Bolzano, CUEN, 1995.

Previous Publications by JURIX

Conference Proceedings

- *Legal Knowledge and Information Systems: The Seventeenth Annual Conference*
  JURIX 2004 (Berlin)
  T. Gordon (ed.)
  ISBN 1 58603 492 8 (IOS Press)
- *Legal Knowledge and Information Systems: The Sixteenth Annual Conference*
  JURIX 2003 (Utrecht)
  D. Bourcier (ed.)
  ISBN 1 58603 398 0 (IOS Press)
- *Legal Knowledge and Information Systems: The Fifteenth Annual Conference*
  JURIX 2002 (London)
  Bench-Capon, T., Daskalopulu, A., Winkels, R. (eds.)
  ISBN 1 58603 299 2 (IOS Press)
  ISBN 4 274 90564 0 C3055 (Ohmsha)
- *Legal Knowledge and Information Systems: The Fourteenth Annual Conference*
  JURIX 2001 (Amsterdam)
  Verheij, B., Lodder, A.R., Loui, R.P., Muntjewerff, A.J. (eds.)
  ISBN 1 58603 201 1 (IOS Press)
  ISBN 4 274 90488 1 C3000 (Ohmsha)
- *Legal Knowledge and Information Systems: The Thirteenth Annual Conference*
  JURIX 2000 (Twente)
  Breuker, J.A., Leenes, R., Winkels, R. (eds.)
  ISBN 1 58603 144 9 (IOS Press)
  ISBN 4 274 90416 4 C3000 (Ohmsha)
- *Legal Knowledge Based Systems: The Twelfth Conference*
  JURIX 1999 (Leuven)
  Herik, H. Jaap van den, Marie-Francine Moens, Jon Bing, Bea van Buggenhout, John Zeleznikow, Carolus Grütters (eds.)
  ISBN 90 71478 64 5
- *Legal Knowledge Based Systems: The Eleventh Conference*
  JURIX 1998 (Groningen)
  Hage, J.C., T.J.M. Bench-Capon, A.W. Koers, C.N.J. de Vey Mestdagh, C.A.F.M. Grütters (eds.)
  ISBN 90 71478 58 0
- *Legal Knowledge Based Systems: The Tenth Conference*
  JURIX 1997 (Amsterdam)
  Oskamp A., R.V. De Mulder, C. van Noortwijk, C.A.F.M. Grütters, K. Ashley, T. Gordon (eds.)
  ISBN 90 71478 54 8
- *Legal Knowledge Based Systems: Foundations of Legal Knowledge Systems*
  JURIX 1996 (9th conference Tilburg)
  Kralingen R.W., H.J. van den Herik, J.E.J. Prins, M. Sergot, J. Zeleznikow (eds.)
  ISBN 90 361 9657 4
- *Legal Knowledge Based Systems: Telecommunications and AI & and Law*
  JURIX 1995 (8th conference Maastricht)

Hage J.C., T.J.M. Bench-Capon, M.J. Cohen, H.J. van den Herik (eds.)
ISBN 90 5458 252 9

- *Legal Knowledge Based Systems: The Relation with Legal Theory*
  JURIX 1994 (7th conference Amsterdam)
  Prakken H., A.J. Muntjewerff, A. Soeteman (eds.)
  ISBN 90 5458 190 5
- *Legal Knowledge Based Systems: Intelligent Tools for Drafting Legislation, Computer-Supported Comparison of Law*
  JURIX 1993 (6th conference Enschede)
  Svensson J.S., J.G.J. Wassink, B. van Buggenhout (eds.)
  ISBN 90 5458 089 5
- *Legal Knowledge Based Systems: Information Technology and Law*
  JURIX 1992 (5th conference The Hague)
  Grütters C.A.F.M., J.A. Breuker, H.J. van den Herik, A.H.J. Schmidt, C.N.J de Vey Mestdagh (eds.)
  ISBN 90 5458 0341 3
- *Legal Knowledge Based Systems: Model-Based Legal Reasoning*
  JURIX 1991 (4th conference Rotterdam)
  Breuker J.A., R.V. de Mulder, J.C. Hage (eds.)
  ISBN 90 6040 989 2
- *Legal Knowledge Based Systems: Aims for Research and Development*
  JURIX 1990 (3rd conference Leiden)
  Noortwijk C. van, A.H.J. Schmidt, R.G.F. Winkels (eds.)
  ISBN 90 6040 948 5
- *Legal Knowledge Based Systems: An Overview of Criteria for Validation and Practical Use (Exploring the Quality of Applications Resulting from Research Programs in the Netherlands)*
  JURIX 1989 (2nd conference Utrecht)
  Kracht D., C.N.J. de Vey Mestdagh, J.S. Svensson (eds.)
  ISBN 90 6040 917 5
- *RI-paradigmata: Toekomst van Nederlands onderzoek in de rechtsinformatica (Paradigms in Legal informatics: Exploring the Future of Dutch Research Programmes in Legal Informatics)*
  JURIX 1988 (1st conference Amsterdam)
  (In Dutch with summaries in English)
  Berg P. van den, C.A.F.M. Grütters, O.W.M. Kamstra, T. van Willigenburg (eds.)
  ISBN 90 6040 886 1

Reports

- *Informatietechnologie en Recht*
  (Rapport van de tijdelijke adviescommissie Informatietechnologie en Recht met betrekking tot de opzet van onderzoek op het gebied van informatietechnologie en recht)
  1991 (In Dutch)
  Franken H., J.C. Arnbak, J.M.A. Berkvens, B.K. Brussaard, A.W. Koers, I.Th.M. Snellen, A.H.J. Schmidt (red.)
  ISBN 90 6040 966 3
- *Information Technology and the Law*
  1991 (English version)
  Franken H., J.C. Arnbak, J.M.A. Berkvens, B.K. Brussaard, A.W. Koers, I.Th.M. Snellen, A.H.J. Schmidt (eds.)
  ISBN 90 6040 987 6

Contact Information on JURIX

Those interested in the activities of JURIX may contact

Radboud Winkels	President of JURIX
Address:	Universiteit van Amsterdam Faculty of Law P.O. Box 1030 NL-1000 BA Amsterdam The Netherlands
Phone:	+31 20 525 3485
Fax:	+31 20 525 3495
E-mail:	winkels@lri.jur.uva.nl
Jurix Website:	http://www.jurix.nl

This page intentionally left blank

Author Index

Aouladomar, F. 81
Ashley, K.D. 107
Atkinson, K. 1
Bench-Capon, T. 1
Biagioli, C. 103
Boer, A. 25
Boschi, L. 71
Cynamon, B.Z. 59
Delgado, J. 137
Dobrov, B. 109
Francesconi, E. 103
García, R. 137
Giblin, C. 37
Governatori, G. 13
Grabmair, M. 107
Hogben, G. 127
Horacek, H. 69
Jacobs, A. 101
Keppens, J. 49
Liu, A.Y. 37
Lodder, A.R. 115
Looks, M. 59, 111
Loui, R.P. 59, 111
Loukachevitch, N. 109
Mercatali, P. 71
Moens, M.-F. v
Mommers, L. 147
Müller, S. 37
Palmirani, M. 13
Pfitzmann, B. 37
Pimenta Rodrigues, I. 91
Quaresma, P. 91
Riveret, R. 13
Romano, F. 71
Rotolo, A. 13
Sartor, G. 13
Schafer, B. 49
Spinicci, E. 71
Spyns, P. v, 127
Stuckenschmidt, H. 115
van Engers, T. 25, 101
van Gog, R. 101
van Harmelen, F. 115
van Laarschot, R. 115
van Steenbergen, W. 115
Vincent, A. 105
Voermans, W. 147
Walton, D. 113
Winkels, R. 25
Wintgens, L.J. 23
Zeleznikow, J. 105
Zhou, X. 37

This page intentionally left blank

This page intentionally left blank